

OPTIMASI PENGIRIMAN BARANG MENGGUNAKAN ALGORITME GENETIKA DENGAN DATA SINTESIS

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aulia Dinia

NIM: 145150200111075



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**

PENGESAHAN

OPTIMASI PENGIRIMAN BARANG MENGGUNAKAN ALGORITME GENETIKA DENGAN DATA SINTESIS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Aulia Dinia
NIM: 145150200111075

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2


Imam Chonssodin, S.Si, M.Kom

NIK: 201201 850719 1 001


Edy Santoso, S.Si, M.Kom

NIP: 19740414 200312 1 004



Mengetahui

Ketua Jurusan Teknik Informatika


Achmad Basuki, S.T., M.MG., Ph.D.

NIP: 19741118 200312 1 002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah dikripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juli 2021



Aulia Dinia

NIM: 145150200111075

PRAKATA

Puji syukur penulis panjatkan atas ke hadirat Allah SWT yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Optimasi Pengiriman Barang Menggunakan Algoritme Genetika Dengan Data Sintesis”. Dalam kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah membimbing dan membantu penulis dalam penyusunan laporan penelitian ini, antara lain:

1. Imam Cholissodin, S.Si., M.Kom., selaku Dosen Pembimbing I yang telah baik hati dan sabar membimbing dalam menyelesaikan skripsi ini.
2. Edy Santoso, S.Si., M.Kom., selaku Dosen Pembimbing II yang dengan ikhlas membimbing dan mengarahkan dalam menyelesaikan penyusunan skripsi ini.
3. Adhitya Bhawiyuga, S.Kom., M.Sc., selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
4. Achmad Basuki, S.T., M.MG., Ph.D., selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
5. Ari Kusyanti, S.T., M.Sc., selaku Dosen Pembimbing Akademik yang telah membimbing selama menempuh perkuliahan.
6. Terutama kedua Orang Tua dan Kakak yang selalu memberikan do’a, motivasi, dan selalu mendukung baik moril dan material.
7. Nadya Silvyani, Yuniar Siska, Eka Miya, Olivia Bonita, dan Albert yang telah memberikan banyak bantuan dalam proses perkuliahan dan pengerjaan skripsi ini.
8. Dan seluruh pihak-pihak lainnya yang telah mendukung. Tidak dapat disebutkan satu persatu.

Penulis menyadari penelitian ini masih memiliki kekurangan, untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun. Semoga penelitian ini dapat digunakan dengan baik dan bermanfaat dalam membantu penelitian selanjutnya.

Malang, 26 Juli 2021

Penulis

auliadina96@gmail.com

ABSTRAK

Aulia Dinia, Optimasi Pengiriman Barang Menggunakan Algoritme Genetika dengan Data Sintesis

Pembimbing: Imam Cholissodin, S.Si., M.Kom dan Edy Santoso, S.Si., M.Kom

Pengiriman barang adalah mempersiapkan pengiriman barang dari Gudang ke tempat tujuan yang telah disesuaikan dengan dokumen pemesanan dan pengiriman serta dalam kondisi yang sesuai dengan persyaratan penanganan barangnya. Permasalahan yang sering terjadi adalah lokasi tujuan yang ditempuh memiliki lokasi yang berbeda-beda, barang yang dikirim tidak sampai ke tempat tujuan, kurir yang tidak mengerti jalur pengantaran barang, dan kurangnya dalam menentukan jarak dari lokasi satu dengan lokasi yang lain. Maka dari itu, diperlukan optimasi pengiriman barang sehingga pengiriman dapat dilakukan dengan lebih efisien. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan ini adalah Algoritme Genetika. Algoritme Genetika merupakan metode komputasi yang digunakan untuk memilih solusi yang sesuai dengan kriteria tanpa harus memeriksa pada seluruh kemungkinan solusi yang ada. Pada penelitian ini, dilakukan analisis global dengan melakukan perbandingan hasil perhitungan manualisasi dengan hasil aktual sistem. Dimana perhitungan manualisasi merupakan hasil pengiriman yang dilakukan oleh JNE tanpa bantuan sistem. Didapatkan selisih *fitness* antara hasil manualisasi pengiriman JNE dengan hasil rekomendasi sistem adalah 0,007950764. Dengan hasil rekomendasi sistem yang mendapatkan *fitness* 0,1083108 lebih baik daripada hasil manualisasi yang menghasilkan *fitness* sebesar 0,1003060036.

Kata kunci: Optimasi, Pengiriman barang, Algoritme Genetika

ABSTRACT

Aulia Dinia, *Optimization of Goods Delivery Using Genetic Algorithm with Synthetic Data*

Supervisors: Imam Cholissodin, S.Si., M.Kom and Edy Santoso, S.Si., M.Kom

Delivery of goods is preparing the delivery of goods from the warehouse to the destination that has been adjusted to the order and delivery documents and in conditions that are in accordance with the requirements for handling the goods. Problems that often occur are the destination locations that are taken have different locations, the goods sent do not reach their destination, couriers who do not understand the delivery route, and lack of determining the distance from one location to another. Therefore, it is necessary to optimize the delivery of goods so that delivery can be carried out more efficiently. One method that can be used to solve this problem is genetic algorithm. Genetic algorithm is a computational method used to select a solution that fits the criteria without having to examine all possible solutions. In this study, a global analysis was carried out by comparing the results of manual calculations with the actual results of the system. Where the manual calculation is the result of deliveries made by JNE without the help of the system. The difference in fitness between the results of manualization of JNE delivery and the results of the system recommendation is 0.007950764. With the results of the recommendation system that gets a fitness of 0.1083108 is better than the results of manualization which produces a fitness of 0.1003060036.

Keywords: *Optimization, Delivery of Goods, Genetic Algorithm.*

PENGESAHAN ii

PERNYATAAN ORISINALITASiii

PRAKATA.....iv

ABSTRAKABSTRACT.....viDAFTAR ISIvii

DAFTAR TABEL.....X

DAFTAR GAMBARxiDAFTAR LAMPIRAN xiii

BAB 1 PENDAHULUAN..... 1

1.1 Latar Belakang..... 1

1.2 Rumusan Masalah.....	3
--------------------------	---

1.3 Tujuan 31.4 Manfaat..... 31.5 Batasan Masalah 31.6 Sistematika Pembahasan 3

BAB 2 LANDASAN KEPUSTAKAAN 5

2.1	Kajian Pustaka	5
-----	----------------------	---

2.2 Optimasi..... 82.3 *Multiple Travelling Salesman Problem (M-TSP)* 8

2.4 Pengiriman Barang

2.5 Jasa Pengiriman Barang 9

2.6 Algoritme Genetika..... 10

2.6.1 Langkah-Langkah Untuk Menyelesaikan Masalah Dalam Algoritma Genetika 11

2.6.2 Siklus Algoritme Genetika..... 12

BAB 3 METODOLOGI PENELITIAN 16

3.1	Tipe Penelitian	16
3.2	Strategi Penelitian.....	16
3.3	Partisipan Penelitian.....	17
3.4	Lokasi Penelitian.....	18
3.5	Teknik Pengumpulan Data.....	18
3.6	Data Penelitian.....	18
3.7	Teknik Analisis Data	19
3.8	Implementasi Algoritme	19
3.9	Peralatan Pendukung.....	19
3.10	Kesimpulan Dan Saran	20
BAB 4 PERANCANGAN.....		21
4.1	Deskripsi Masalah	21
4.2	Siklus Algoritme Genetika.....	22
4.3	Siklus Penyelesaian Masalah Menggunakan Algoritme Genetika ...	25
4.3.1	Representasi Kromosom.....	25
4.3.2	Inisialisasi Populasi Awal.....	26
4.3.3	Reproduksi	30
4.3.4	Perhitungan Nilai <i>Fitness</i>	41
4.3.5	Evaluasi dan Seleksi	45
4.4	Perancangan Antarmuka	53
4.4.1	Perancangan Halaman Data	53
4.4.2	Perancangan Halaman Parameter.....	54
4.4.3	Perancangan Halaman Algoritme Genetika.....	56
4.5	Perancangan Pengujian	57
4.5.1	Uji Coba Ukuran Populasi	57
4.5.2	Uji Coba Banyak Generasi.....	58
4.5.3	Uji Coba Kombinasi Cr & Mr.....	59
4.6	Uji Coba Analisis Global	60
BAB 5 IMPLEMENTASI.....		62

5.1	Implementasi Program	62
5.2	Proses Pengambilan Data	62
5.3	Proses <i>Input</i> Parameter	64
5.3.1	Proses memasukkan Data Pengirim	67
5.3.2	Proses Memasukkan Data Kurir	68
5.3.3	Proses Memasukkan Data Parameter Algoritme Genetika	69
5.4	Proses Inisialisasi Populasi Awal	71
5.5	Proses Reproduksi	73
5.5.1	Proses Crossover	73
5.5.2	Proses Mutasi	76
5.6	Perhitungan Total Jarak Dan Fitness	77
5.7	Proses Evaluasi Dan Seleksi	79
5.8	Implementasi Perancangan Antarmuka	82
5.8.1	Tampilan Halaman Data	82
5.8.2	Tampilan Halaman Parameter	82
5.8.3	Tampilan Halaman Algoritme Genetika	83
BAB 6 HASIL DAN PEMBAHASAN		84
6.1	Uji Coba Ukuran Populasi Dan Pembahasan Hasil	84
6.2	Uji Coba Banyak Generasi Dan Pembahasan Hasil	85
6.3	Uji Coba Kombinasi Cr & Mr Dan Pembahasan Hasil	87
6.2	Analisis Global	89
BAB 7 PENUTUP		92
7.1	Kesimpulan	92
7.2	Saran	93
DAFTAR REFERENSI		94

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	6
Tabel 4.1 Data Jarak Tiap Penerima	21
Tabel 4.2 Representasi Kromosome	25
Tabel 4.3 Inisialisasi Populasi Awal	29
Tabel 4.4 Proses Crossover	35
Tabel 4.5 Proses Mutasi	41
Tabel 4.6 Perhitungan Nilai Fitness	45
Tabel 4.7 Hasil Evaluasi	47
Tabel 4.8 Hasil Seleksi	50
Tabel 4.9 Hasil Proses Pengurutan Individu	53
Tabel 4.10 Perancangan Uji Coba Ukuran Populasi	58
Tabel 4.11 Perancangan Uji Coba Banyak Generasi	58
Tabel 4.12 Perancangan Uji Coba Kombinasi Cr & Mr	59
Tabel 4.13 Perancangan Uji Coba Perhitungan Pengiriman Manualisasi	60
Tabel 4.14 Perancangan Uji Coba Sistem	60
Tabel 4.15 Perancangan Uji Coba Perbandingan Rekomendasi Sistem Dan Manualisasi	60
Tabel 6.1 Uji Coba Ukuran Populasi	84
Tabel 6.2 Uji Coba Banyak Generasi	86
Tabel 6.3 Uji Coba Kombinasi Cr & Mr	88
Tabel 6.4 Uji Coba Perhitungan hasil Pengiriman Manualisasi	89
Tabel 6.5 Uji Coba Rekomendasi Sistem	90
Tabel 6.6 Uji Coba Perbandingan Rekomendasi Sistem Dan Manualisasi	90

DAFTAR GAMBAR

Gambar 2.1 Kantor JNE Lamongan	10
Gambar 3.1 Diagram Blok optimasi Pengiriman barang dengan Algoritme Genetika	17
Gambar 4.1 Diagram Alir Algoritme Genetika	23
Gambar 4.2 Diagram Inisialisasi Populasi Awal	27
Gambar 4.2 Diagram Inisialisasi Populasi Awal (Lanjutan)	28
Gambar 4.3 Diagram Alir Crossover	32
Gambar 4.3 Diagram Alir Crossover (Lanjutan)	33
Gambar 4.4 Diagram Proses Crossover	34
Gambar 4.5 Diagram Proses Mutasi	37
Gambar 4.6 Diagram Alir Mutasi	38
Gambar 4.6 Diagram Alir Mutasi (Lanjutan)	39
Gambar 4.7 Diagram Alir Hitung Nilai Fitness	42
Gambar 4.7 Diagram Alir Hitung Nilai Fitness (Lanjutan)	43
Gambar 4.8 Diagram Alir Evaluasi	46
Gambar 4.9 Diagram Alir Seleksi	48
Gambar 4.9 Diagram Alir Seleksi (Lanjutan)	49
Gambar 4.10 Diagram Alir Pengurutan Individu	51
Gambar 4.10 Diagram Alir Pengurutan Individu (Lanjutan)	52
Gambar 4.11 Perancangan Antarmuka Halaman Data	54
Gambar 4.12 Perancangan Antarmuka Halaman Parameter	55
Gambar 4.13 Perancangan Antarmuka Halaman Algoritme Genetika	56
Gambar 5.1 Hasil Tampilan Halaman Data	82
Gambar 5.2 Hasil Tampilan Halaman Parameter	83
Gambar 5.3 Hasil Tampilan Halaman Algoritme Genetika	83
Gambar 6.1 Hasil Uji Coba Ukuran Populasi	85
Gambar 6.2 Hasil Uji Coba Banyaknya Generasi	87

Gambar 6.3 Hasil Uji Coba Kombinasi Cr & Mr 88

Gambar 6.4 Hasil Uji Coba Perbandingan Sistem Rekomendasi Dan Manualisasi 91



DAFTAR LAMPIRAN

LAMPIRAN A DATA ALAMAT PENGIRIM.....	98
LAMPIRAN B DATA JARAK TIAP PENERIMA	99
LAMPIRAN C PERHITUNGAN NILAI FITNESS.....	100
LAMPIRAN D UJI COBA UKURAN POPULASI.....	101
LAMPIRAN E UJI COBA BANYAK GENERASI.....	102
LAMPIRAN F UJI COBA KOMBINASI CR & MR	103



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya zaman dan persaingan dalam perusahaan yang semakin ketat, membuat para pengusaha dituntut harus lebih efisiensi supaya mendapatkan keuntungan yang maksimal (Gazali & Manik, 2010). Pengiriman barang adalah mempersiapkan pengiriman barang dari Gudang ke tempat tujuan yang telah disesuaikan dengan dokumen pemesanan dan pengiriman serta dalam kondisi yang sesuai dengan persyaratan penanganan barangnya (Wibowo, dkk., 2019). Di saat seperti ini, banyak masyarakat menggunakan media online untuk jual beli online. Para online store harus pintar membuat promosi setiap bulannya supaya menarik perhatian pembeli. Transaksi jual beli online ini dilakukan oleh pembeli dengan memesan barang secara online pada online store tersebut, kemudian barang yang telah dipesan akan dikirimkan dengan menggunakan jasa pengiriman barang. Dengan tingginya jumlah pembeli barang online mengakibatkan meningkatnya akan jasa pengiriman barang. Beberapa permasalahan yang sering terjadi saat pengiriman barang adalah lokasi tujuan yang ditempuh memiliki lokasi yang berbeda-beda, barang yang dikirim tidak sampai ke tempat tujuan, kurir yang tidak mengerti jalur pengantaran barang, dan kurangnya dalam menentukan jarak dari lokasi satu dengan lokasi yang lain.

Penelitian pertama yang dilakukan oleh (Hanafi, dkk., 2018) untuk menyelesaikan permasalahan pengiriman barang Kantor Pos Lumajang. Tujuan dari penelitian ini adalah untuk mendapatkan solusi yang optimum. Para perusahaan *online* mengeluarkan promosi yang tidak ada henti-hentinya kepada pembeli yang mengakibatkan meningkatkan jumlah pengiriman barang. Para jasa pengiriman barang berusaha untuk bisa melayani dengan baik dan juga harus meminimalkan biaya pengiriman barang. Perusahaan Kantor Pos di Lumajang mempunyai 4 seles dengan 16 tujuan pengiriman dengan *Multi Travelling Salesman Problem (M-TSP)*. Hasil pengujian yang didapatkan dari permasalahan ini adalah parameter yang optimal dengan ukuran populasi 200, maksimum generasi 500, kombinasi *Crossover Rate* 0,4, dan *Mutation Rate* 0.6 sehingga didapatkan *fitness* sebesar 0,05288.

Penelitian ke dua yang dilakukan oleh (Suprayogi & Mahmudy, 2015) mengimplemetasikan Algoritme Genetika dalam menyelesaikan permasalahan antar jemput *laundry*. Penelitian tersebut bertujuan untuk memperoleh optimasi rute perjalanan yaitu rekomendasi terbaik untuk jalur yang akan dilalui dan waktu perjalanan yang cepat dan efisien namun tetap dapat menjangkau semua pelanggannya. Perhitungan rute tercepat dirasa penting karena mengharuskan

barang diantar dengan tepat waktu dan tidak ada keluhan pelanggan mengenai keterlambatan pengantaran. Untuk mendapatkan solusi yang terbaik atas permasalahan diatas, maka digunakan beberapa kombinasi probabilitas *crossover* dan mutasi serta ukuran populasi dan ukuran generasi. Nilai-nilai parameter yang memungkinkan untuk didapatkan solusi yaitu untuk melayani semua pelanggan dengan time windows masing-masing. *Dataset* yang digunakan juga dapat mempengaruhi nilai optimal pada algoritme genetika. Hasil pengujian pada permasalahan ini adalah memiliki probabilitas *crossover* optimal yaitu sebesar 0,4 dan probabilitas mutasinya sebesar 0,6 dengan ukuran generasi sebesar 2000.

Penelitian ke tiga yang di lakukan oleh (Arif, dkk., 2019) untuk menyelesaikan permasalahan jalur distribusi barang. Tujuan dari penelitian ini adalah untuk menentukan jalur distribusi barang terbaik supaya waktu dan biaya perjalanan lebih optimal. Pada penentuan jalur ini lebih memfokuskan pendistribusian barang pada agen-agen yang ada di daerah Jember. Pendistribusian barang hanya di daerah Jember saja dengan 12 perjalanan dalam satu jalur pendistribusian barang. Hasil pengujian dari permasalahan ini adalah hasil penentuan jalur pada *fitness* gen pertama didapatkan *fitness* tertinggi 7,4 dan *fitness* terendah 5,6, pada *fitness* gen kedua didapatkan *fitness* tertinggi 9,3 dan *fitness* terendah 5,6.

Penelitian ke empat yang di lakukan oleh (Pratama, dkk., 2017) untuk menyelesaikan permasalahan dalam pencarian jalur pengantaran barang dagang *online* dengan cara pemilihan jalur tujuan pengantaran barang yang akan dilewati secara bersamaan oleh 4 kurir dengan transportasi yang berbeda-beda dan kapasitas angkut tiap transportasi yang telah ditentukan. Transportasi yang di pakai untuk pengantaran barang yaitu mobil dan motor. Pengujian yang dilakukannya 10 kali percobaan dengan parameter masukan penghentian generasi untuk maksimal generasu sebanyak 1000 generasi, batas nilai kecocokan berturut-turut pada generasi sebelumnya sebanyak 50 generasi. Hasil pengujian pada permasalahan ini adalah nilai kecocokan terbesar pada pengujian ke-1 sebesar 702,2 dalam waktu proses selama 2,5111 menit dengan jumlah generasi sebanyak 357 generasi.

Dengan beberapa penelitian tersebut maka metode yang dapat digunakan untuk menyelesaikan permasalahan pengiriman barang ini adalah algoritme genetika. Algoritme Genetika merupakan metode komputasi yang digunakan untuk memilih solusi yang sesuai dengan kriteria tanpa harus memeriksa pada seluruh kemungkinan solusi yang ada (Hasyim, dkk., 2017).

Berdasarkan penjelasan beberapa penelitian diatas, maka mengajukan penelitian ini dengan judul "Optimasi Pengiriman Barang Menggunakan Algoritme Genetika dengan Data sintesis". Pada penelitian ini bertujuan untuk mengetahui pengiriman barang yang paling efektif dan cepat.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang pada penelitian diatas, maka dapat diperoleh rumusan masalahnya sebagai berikut:

1. Bagaimana hasil parameter Algoritme Genetika yang didapat dalam optimasi pengiriman barang dengan algoritma genetika?
2. Bagaimana tingkat keberhasilan yang diperoleh dari pengiriman barang dengan algoritme genetika?

1.3 Tujuan

Tujuan pada penelitian ini adalah sebagai berikut:

1. Untuk mengetahui hasil parameter algoritme genetika yang didapat dari optimasi pengiriman barang dengan algoritme genetika.
2. Untuk mengetahui tingkat keberhasilan yang diperoleh dari permasalahan dalam pengiriman barang dengan algoritme genetika.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Membantu kurir dalam pengiriman barang supaya barang sampai ke tempat tujuan.
2. Untuk mengetahui jarak dari lokasi satu ke lokasi yang lainnya yang paling optimal.
3. Sebagai referensi untuk penelitian berikutnya.

1.5 Batasan Masalah

Batasan masalah pada penelitian adalah sebagai berikut:

1. Titik awal pengiriman barang yaitu dari Kantor ekspedisi JNE.
2. Kendaraan yang digunakan untuk pengiriman barang yaitu motor.
3. Hanya fokus pada pengiriman barang saja tanpa memperhatikan volume barang yang dibawah oleh setiap kurir.
4. Data jarak di peroleh dari *Google Maps*.

1.6 Sistematika Pembahasan

Sistematika penyusunan laporan ini meliputi beberapa Bab, yaitu sebagai berikut:

BAB 1: PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, manfaat, tujuan, batasan masalah, dan sistematika pembahasan.

BAB 2: LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai dasar teori atau referensi yang memiliki keterkaitan dengan optimasi dan Algoritme Genetika. Pada kajian pustaka terdapat beberapa sub bab dapat menunjang dalam proses pembuatan skripsi:

1. Optimasi
2. Pengiriman Barang
3. Multiple Travelling Salesman Problem(M-TSP)
4. Algoritme Genetika

BAB 3: METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi dan langkah-langkah kerja yang dilakukan pada proses perancangan dan implemetasi adalah tipe penelitian, strategi penelitian, partisipan penelitian, lokasi penelitian, teknik pengumpulan data, data penelitian, teknik analisis data, implemetasi algoritma, jadwal penelitian pada pengiriman barang menggunakan Algoritme Genetika.

BAB 4: PERANCANGAN

Bab ini menjelaskan tentang proses kebutuhan dan susunan kerja dalam perancangan optimasi pengiriman barang menggunakan algoritma genetika, mulai dari perancangan model sampai dengan antar muka sistem.

BAB 5: IMPLEMENTASI

Bab ini berisi ulasan yang membahas tahapan-tahapan pada penerapan dari perancangan yang tersedia. Tahapan pada implementasi antara lain spesifikasi dari sistem, penerapan algortima, dan pengimplementasian antarmuka dalam penelitian pada pengiriman barang menggunakan Algoritme Genetika.

BAB 6: HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai pembahasan hasil pengujian sistem pada optimasi pengiriman barang menggunakan Algoritme Genetika.

BAB 7: PENUTUP

Bab ini menjelaskan mengenai kesimpulan dan saran terhadap sistem optimasi pengiriman barang menggunakan Algoritme Genetika.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian pertama yang dilakukan oleh (Arif, dkk., 2019) untuk menyelesaikan permasalahan jalur distribusi barang. Tujuan dari penelitian ini adalah untuk menentukan jalur distribusi barang terbaik supaya waktu dan biaya perjalanan lebih optimal. Pada penentuan jalur ini lebih memfokuskan pendistribusian barang pada agen-agen yang ada di daerah jember. Pendistribusian barang hanya di daerah Jember saja dengan input 12 tujuan dalam satu jalur pendistribusian barang. Metode algoritma genetika bisa digunakan perangkat lunak. Dalam perangkat lunak ini ada beberapa inputan yang dibutuhkan yaitu pengiriman barang pada tujuan kota sebagai kromosom awal, jumlah generasi, probabilitas, *crossover* dan probabilitas mutasi. Penelitian yang digunakan yaitu menggunakan algoritma genetika. Berdasarkan 5 kombinasi jalan yang digunakan sebagai pengujian untuk rute pengiriman *chip*. Hasil pengujian dari permasalahan ini adalah hasil penentuan jalur pada *fitness* gen pertama didapatkan *fitness* tertinggi 7,4 dan *fitness* terendah 5,6, pada *fitness* gen kedua didapatkan *fitness* tertinggi 9,3 dan *fitness* terendah 5,6.

Penelitian selanjutnya yang dilakukan oleh (Pratama, dkk., 2017) yang menerapkan algoritme genetika dalam pengantaran barang dagangan *online*. Tujuan dari penelitian ini adalah untuk memberikan rekomendasi kurir dan jalur tujuan pengiriman barang yang dilewati sesuai berdasarkan jumlah angkut. Pada penelitian ini, setiap kurir yang sedang beroperasi menggunakan kendaraan memiliki jumlah angkut yang sudah ditentukan. Pengiriman barang bisa dilakukan oleh jasa ekspedisi yang biasanya mengantarkan barang pada satu wilayah supaya barang yang dikirim tepat waktu dan efisien. Pemberangkatan tiap kurir dimulai dari titik awal pemberangkatan dengan waktu yang bersamaan, supaya semua tujuan pengiriman barang bisa dilewati oleh kurir. Kurir harus mengirimkan barang ke tempat tujuan hanya satu kali pengiriman dan tidak ada tujuan yang sama dalam pengiriman lebih dari 1 kurir. Algoritme Genetika bisa diterapkan dalam optimalisasi jadwal, rute, dan *space*. Pada tahap algoritme genetika dimulai dengan pembangkitan awal secara acak. Setiap individu di evaluasi secara acak, kemudian diseleksi menjadi setengah dari populasi awal. Individu yang sudah terpilih akan dilakukan proses persilangan dengan menyilangkan antar dua individu, kemudian akan memperoleh individu baru. Setelah dilakukan proses persilangan, selanjutnya dilakukan proses mutasi dengan menukarkan posisi dua gen yang telah dipilih secara acak. Pada proses evaluasi ini

yaitu pencocokan, seleksi, persilangan, dan mutasi ini terus dilakukan secara berulang-ulang sampai memenuhi kriteria dan generasi sudah terpenuhi. Sehingga, individu akhir menjadi individu yang telah memenuhi kriteria sebagai solusi yang optimal. Didapatkan solusi dalam penelitian ini menghasilkan paling tinggi sebesar 702.2, dengan waktu 2,5111 menit dengan jumlah generasi sebanyak 357.

Tabel 2.1 Kajian Pustaka

No	Pustaka	Objek & Metode	Output
1	(Hasyim, dkk., 2017)	Optimalisasi Rute Objek Wisata di Bandung Raya Menggunakan Algoritme Genetika	Hasil pengujian yaitu nilai kecocokan yang paling tinggi sebesar 910.2431 dari 117 generasi dengan waktu 102 menit pada pengujian ke satu Algoritma Genetika ke 1.
2	(Rizki, dkk., 2017)	Optimasi Multi Travelling (M-TSP) Untuk Distribusi Produk Pada Home Industri Tekstil Dengan Algoritme Genetika	Hasil pengujian yang diperoleh yaitu jumlah populasi sebesar 120, jumlah generasi sebesar 800, nilai kombinasi Cr sebesar 0,4 dan nilai Mr sebesar 0,6.
3	(Hanafi, dkk., 2018)	Penyelesaian <i>Multi Travelling Salesman Problem</i> (M-TSP) Dengan Menggunakan Algoritme Genetika: Studi Kasus Pendistribusian Barang di Kantor Pos Lumajang	Hasil pengujian yang didapatkan adalah nilai populasi sebesar 200, nilai maksimum generasi 500, nilai <i>Crossover Rate</i> 0,4, dan nilai <i>Mutation Rate</i> sebesar 0,6. Hasil fitness sebesar 0,05288.
4	(Pratama, dkk., 2017)	Optimalisasi Pengantaran Barang dalam Perdagangan Online Menggunakan Algoritma Genetika	Hasil pengujian yang di dapatkan adalah nilai kecocokan paling tinggi sebesar 702.2, dengan waktu 2,5111 menit, jumlah generasi sebanyak 357.
5	(Suprayogi & Mahmudy, 2015)	Penerapan Algoritme Genetika Travelling Salesman Problem with Time Windows. Studi	Hasil pengujian yaitu generasi yang optimal 2000 generasi, nilai <i>crossover</i> sebesar 0,4, dan nilai mutasi sebesar 0,6.

		Kasus: Rute Antar Jemput Laundry	
6	(Supriana, 2017)	Implementasi Algoritme Genetika dalam Penentuan Rute Terbaik Pendistribusian BBM Pada PT Burung Laut	Hasil pengujian yang didapatkan yaitu <i>fitness</i> terbesar yang berada pada pengujian ke-5, <i>fitness</i> yang didapat sebesar 2.0606, <i>fitness</i> terbesar yang menghasilkan rute terpendek sebesar 4.853 kilometer.
7	(Ahn & Ramakrishna, 2002)	<i>A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations</i>	Hasil pengujian didapatkan menunjukkan bahwa prediksi persamaan yang di dapatkan di wilayah operasional (praktis) yang lebih baik dari 0,1 probabilitas kegagalan rute (90% optimalitas rute)
8	(Baskara Joni & Nurcahyani, 2012)	Penentuan Jarak Terpendek Pada Jalur Distribusi Barang Di Pulau Jawa Dengan Menggunakan Algoritme Genetika	Hasil pengujian yang didapatkan adalah rute pengiriman dengan nilai <i>fitness</i> sebesar 762.
9	(Tahyudin & Susanti, 2015)	Pencarian Rute Terbaik pada Objek Wisata di Kabupaten Banyumas Menggunakan Algoritme Genetika Metode TSP (<i>Travelling Salesman Problem</i>)	Hasil yang didapatkan dari penelitian ini yaitu panjang jalur terbaik sebesar 0,878 unit kartesian, ukuran populasi banyak 25, dan probabilitas mutasi sebesar 0,005.
10	(Basuki, 2017)	Penentuan Rute Optimum Produk PT Indmira Berdasarkan Jarak	Hasil pengujian ini adalah rute yang bisa menghemat jarak dalam 1 tahun yaitu 20,8 kilometer. BBM yang di gunakan pada rute usulan adalah 160.94 liter. biaya rute usulan adalah Rp. 1.190.956.
11	(Natalia, dkk., 2019)	Optimasi Pengiriman Sapi Potong pada Jaringan Pelayaran	Hasil pengujian ini adalah usulan jarak optimal scenario 1 total biaya sebesar Rp 443.306.533.26 dengan

	Kapal Khusus Ternak dengan Algoritme Genetika	jumlah sapi sebanyak 1374 ekor sapi, sedangkan untuk scenario 2 total biaya sebesar Rp 1.321.505.850.62 dengan jumlah sapi sebanyak 3416 ekor sapi.
--	---	---

Berdasarkan hasil penelitian yang tercantum pada Tabel 2.1, maka dapat diperoleh kesimpulan bahwa algoritme genetika merupakan salah satu metode yang sering digunakan untuk penentuan rute. Pada setiap pengujian yang dilakukan pada jurnal-jurnal diatas didapatkan rute terpendek ditemukan dengan nilai *fitness* terbesar. Proses pembangunan rute menggunakan algoritma genetika dapat dilakukan secara optimal. Penerapan algoritma genetika cukup membantu dalam mencari rute terpendek pada kasus yang bermodel TSP.

2.2 Optimasi

Optimasi adalah suatu pencarian hasil terbaik yang bertujuan untuk memperoleh solusi yang mendekati pada suatu permasalahan. Di kehidupan sehari-hari, permasalahan optimasi sangat sering terjadi (Hasibuan & Lusiana, 2015). Teknik optimasi bisa juga untuk memperoleh titik minimum atau titik maksimum yang didapat. Untuk memperoleh keuntungan yang maksimum, dibutuhkan suatu teknik pengolahan data masukan supaya memperoleh hasil berupa *fitness*, yang mana hasil nilai *fitness* dijadikan sebagai acuan untuk memperoleh solusi yang maksimum (Sari & Mahmudy, 2015).

2.3 Multiple Travelling Salesman Problem (M-TSP)

Permasalahan yang sering terjadi pada *Travelling Salesman Problem* adalah bagaimana dalam menentukan rute perjalanan dari tempat satu ke tempat yang lainnya supaya optimal (Permanasari & Salim, 2006). Menurut (Kusrini & Istiyanto, 2007) TSP merupakan permasalahan pada pencarian rute terpendek yang dapat dilewati oleh sales yang ingin mengunjungi ke beberapa tempat supaya tidak berkunjung tempat yang sama lebih dari satu kali. Adapun beberapa contoh yang dapat diselesaikan dengan menggunakan TSP yaitu pencarian rute bis sekolah untuk mengantar siswa, pencarian rute truk untuk mengantarkan parcel, pengambilan tagihan telepon, dst. Pada pengiriman yang berskala besar tidak cukup hanya menggunakan satu kurir dengan banyak yang sangat banyak dan terus bertambah. Pada proses pengiriman barang membutuhkan lebih dari satu kurir. Oleh sebab itu

diperlukan sebuah penyelesaian supaya dapat diselesaikan secara efisien. Pendekatan yang digunakan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan MTSP atau *Multiple Traveling Salesman Problem* karena pada permasalahan pengiriman barang membutuhkan lebih dari satu kurir untuk mengunjungi beberapa tempat dan Kembali ke tempat awal lagi.

2.4 Pengiriman Barang

Pengiriman barang adalah suatu usaha yang bertujuan untuk memberikan pelayanan supaya terlaksananya suatu pengiriman, pengangkutan dan penerimaan barang dengan menggunakan transportasi baik melalui darat (Supriati, dkk., 2017). Proses pengiriman barang merupakan kegiatan yang tidak jauh dari kehidupan kita. Jarak yang jauh begitu juga dengan penyebaran masyarakat yang sangat luas menjadikan salah satu alasan bagi masyarakat untuk memakai jasa pengiriman barang daripada mengantarakan barang itu sendiri yang akan dikirimkan. Masalah pengiriman barang menjadi poin penting bagi perusahaan untuk menyediakan jasa pengiriman barang (Sari, dkk., 2013).

2.5 Jasa Pengiriman Barang

Jasa pengiriman barang menurut definisi umum adalah segala upaya penyelenggaraan atau pelaksanaan baik secara mandiri atau bersama-sama dalam suatu organisasi untuk menghasilkan pelayanan yang efektif dan efisien. Penggunaan jasa pengiriman juga semakin masif karena jarak tempuh yang jauh dan juga karena factor kecepatan waktu dalam pengiriman. Jasa pengiriman cukup membantu untuk pengiriman barang ke seseorang. Mungkin jaraknya memang sangat jauh atau mungkin tidak begitu jauh, yang pasti jasa pengiriman barang ikut serta dalam proses sampainya barang yang dikirim kepada penerima. Salah satu perusahaan dibidang jasa pengiriman barang adalah Jalur Nugraha Ekakurir (JNE) (Triyani, 2017).

JNE (Jalur Nugraha Ekakurir) adalah perusahaan kurir dan logistik terbesar di Indonesia yang melayani pengiriman *express* merupakan layanan reguler sampai Kota dan kabupaten tujuan di seluruh Indonesia. Pada produk pelayanan ini sangat bermacam-macam yang di sertai laporan penyampaian secara otomatis melalui layanan SMS merupakan prosuk andalan JNE saat ini. Walaupun layanan JNE berpengaruh pada faktor lainnya seperti cuaca dan layanan penerbangan, namun dengan adanya jaminan uang Kembali merupakan komitmen JNE untuk memberikan pelayanan terbaik pada pelanggannya (Hutasoit, 2019).

JNE Lamongan merupakan salah satu Kantor cabang pengiriman barang yang berada di Jl. Suanan Drajat No. 332 E, Kaloharjo, Sidoharjo, Kec. Lamongan, Kabupaten Lamongan.



Gambar 2.1 Kantor JNE Lamongan

2.6 Algoritme Genetika

Menurut (Saputro, dkk., 2015) Algoritme genetika dikenalkan oleh John Holland untuk menyelesaikan suatu permasalahan optimasi Pertama kali algoritme genetika di pelajari oleh John Holland pada tahun 1975 di New York Amerika Serikat oleh John Holland di Universitas Michingin dan dikenalkan oleh muridnya yang bernama Goldberg (Lestari, dkk., 2019). Algoritme Genetika juga merupakan tipe algoritme evolusi yang sangat populer. Algoritme Genetika bertumbuh dengan sangat pesat bersamaan dengan perkembangan teknologi informasi. Algoritme Genetika dapat menyelesaikan berbagai masalah dan dapat digunakan dalam bidang lainnya seperti fisika, ekonomi, biologi yang sering menghadapi pada permasalahan optimasi menggunakan permodelan matematika yang kompleks (Mahmudy, 2015).

Algoritme Genetika bisa diimplementasikan pada proses optimasi. Hal ini disebabkan oleh sifat perubahan evolusinya yang alamiah. Algoritma ini hanya dapat memproses perhitungan matematis yang sederhana untuk masalah yang akan diselesaikannya, mencari penyelesaian tanpa memperhatikan proses-proses yang berhubungan dengan masalah yang diselesaikan secara langsung. Selain itu operator-operator evolusi membuat Algoritme Genetika sangat efektif untuk pencarian global (Zukhri, 2014).

Solusi pada masalah tersebut dipetakan menjadi pita kromosom yang disusun dari beberapa gen yang menggambarkan variabel-variabel solusi (Mahmudy, 2015).

2.6.1 Langkah-Langkah Untuk Menyelesaikan Masalah Dalam Algoritme Genetika

Langkah-langkah dalam menyelesaikan permasalahan algoritme genetika (Mahmudy, 2015) secara umum yaitu sebagai berikut:

1. Mulai yaitu menambahkan nilai parameter yang diperlukan, dapat juga dengan cara memasukkan nilai parameter sesuai dengan permasalahan dan nilai parameter Algoritme Genetika.
2. Inisialisasi populasi awal adalah proses pembentukan pita kromosom sebanyak *popsiz*.
3. *Crossover* adalah proses pemindahan silang bagi setiap individu yang sudah dipilih secara acak guna untuk proses reproduksi.
4. *Mutation* adalah proses untuk mentransformasi nilai gen dalam titik tertentu. Langkah ini dilakukan untuk memilih satu induk secara acak lalu dilakukan perubahan pada nilai gen sesuai dengan metodologi yang digunakan yang kemudian akan memperoleh nilai *offspring*.
5. Evaluasi adalah proses untuk menggabungkan antara *parent* dengan child (*offspring*) dan menghitung nilai *fitness* pada setiap individu.
6. Seleksi adalah proses untuk mengurutkan individu sesuai dengan nilai *fitness*, pengurutan ini menggunakan metode seleksi.
7. *Test* yaitu kondisi apabila kriteria sudah terpenuhi maka langkah perhitungan berhenti lalu akan menampilkan solusi dari populasi terakhir.
8. *Loop* yaitu kondisi jika belum memenuhi kriteria maka akan kembali ke nomor 1.

2.6.2 Siklus Algoritme Genetika

2.6.2.1 Representasi Kromosom

Representasi kromosom adalah proses untuk memberikan perwakilan [ada bentuk simbol. Terdapat berbagai macam kromosom. Salah satunya menggunakan representasi permutasi (Mahmudy, 2015). Representasi kromosom adalah mengubah data menjadi gen-gen pembentuk kromosom, dimana gen merupakan bagian dari kromosom. Sebuah variable umumnya dapat diwakilkan oleh satu gen. Supaya dapat diproses melalui Algoritme Genetika, maka proses pengkodean solusi alternatif harus dilakukan sehingga menjadi bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengkodekan informasi yang disimpan di dalam kromosom. (Sinaga, 2019).

Gen dapat direpresentasikan dalam bentuk *bit*, bilangan *real*, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. (Sinaga, 2019). Dengan demikian kromosom dapat direpresentasikan dengan menggunakan:

- 1) String bit : 10011 dst.
- 2) Array bilangan real : 65.65, -67.98, 77.34 dst.
- 3) Elemen permutasi : E2, E10, E5 dst.
- 4) Daftar aturan : R1, R2, R3 dst.
- 5) Struktur lainnya. (Sinaga, 2019).

2.6.2.2 Inisialisasi Populasi Awal

Inisialisasi merupakan suatu individu yang mempunyai gen (*chromosome*) tertentu. Tujuan dari inisialisasi yaitu untuk membangkitkan solusi baru secara *random* yang terdiri atas beberapa *string chromosome* dan diletakkan pada penampungan yang sering disebut populasi (Mahmudy, 2015). Proses inisialisasi ini dilakukan dengan cara memberikan nilai awal gen-gen dengan nilai acak sesuai batasan yang telah ditentukan sebelumnya. (Lubis & Ginting, 2016).

Proses pembangkitan populasi awal adalah sebuah proses pembangkitan beberapa individu secara tidak berurutan atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan,

kemudian dilakukan pembangkitan populasi awal. (Sinaga, 2019). Dalam proses pembangkitan populasi awal terdapat beberapa cara, diantaranya adalah *random generator*, pendekatan tertentu, dan permutasi gen. Pada skripsi ini digunakan teknik pembangkitan populasi berupa *random generator*. Inti dari cara ini adalah melibatkan pembangkitan bilangan *random* untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan. Misalkandalam populasi terdapat 4 individu, maka contoh populasi awal TSP dengan 7 kota (kota1 sampai 7) adalah sebagai berikut:

```

1 2 4 3 6 5 7
5 7 4 6 2 3 1
7 2 6 4 5 1 3
4 6 5 7 3 2 1

```

2.6.2.3 Reproduksi

Reproduksi merupakan suatu proses untuk menghasilkan keturunan dari individu-individu yang terdapat pada populasi. Kumpulan keturunan ini diletakkan dalam proses *offspring*. Dua operator genetika yang digunakan pada proses reproduksi inia adalah tukar silang (*crossover*) dan mutasi (*mutation*). Terdapat banyak metode *crossover* dan *mutation* yang telah dikembangkan oleh beberapa peneliti dan bersifat spesifik pada suatu masalah dan representasi kromosom yang digunakan (Mahmudy, 2015).

Reproduksi ini dilakukan untuk menghasilkan keturunan (*offspring*) dari individu-individu pada populasi. Dalam tahap ini harus ditentukan tingkat *crossover* (*Crossover Rate / Cr*). Nilai ini menyatakan rasio *offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $Cr \times popSize$. Nilai tingkat mutasi (*Mutation Rate / Mr*) juga harus ditentukan. Nilai ini menyatakan rasio *offspring* yang dihasilkan dari proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $Mr \times popSize$. (Mahmudy, 2015).

2.6.2.3.1 Crossover

Crossover adalah proses memilih dua induk (*parent*) dari populasi secara acak. Metode yang digunakan adalah Metode *One-Cut-Point*, yaitu memilih secara acak pada satu titik potong dan menukarkan pada bagian kanan dari tiap induk (*parent*)

untuk menghasilkan *offspring* (Mahmudy, 2015). Pindah silang (*Crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan keturunan baru dalam ruang pencarian yang siap diuji. Operasi ini tidak selalu dilakukan pada setiap individu yang ada. Individu dipilih secara acak untuk dilakukan *crossover* dengan PC (Probabilitas *crossover*) antara 1 s/d 100. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan (Sinaga, 2019).

Prinsip dari pindah silang ini adalah melakukan operasi (pertukaran aritmatika) pada gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang ditentukan. Dalam tahap ini harus ditentukan tingkat *crossover* (*Crossover Rate / Cr*). Nilai ini menyatakan rasio *offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $Cr \times popSize$ (Mahmudy, 2015).

2.6.2.3.2 Mutation

Menurut (Lukas, dkk., 2005) mutasi adalah proses menukarkan nilai gen secara acak. Mutasi dilakukan dengan cara menggeser nilai gen terhadap gen yang sudah terpilih untuk dimutasikan (Sofwan, dkk., 2008). Nilai tingkat mutasi (*Mutation Rate / Mr*) juga harus ditentukan. Nilai ini menyatakan rasio *offspring* yang dihasilkan dari proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $Mr \times popSize$ (Mahmudy, 2015).

Mutasi ialah proses untuk mengubah nilai dalam suatu kromosom dari beberapa gen atau bahkan dari satu gen. Tujuan proses mutasi ini yaitu untuk memperoleh beberapa kromosom baru sebagai kandidat dalam generasi berikutnya untuk mendapatkan nilai fitness yang lebih baik, dan secara bertahap akan mendapatkan solusi optimum sesuai yang diharapkan. Penekanan selektif memiliki peranan penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi (Sinaga, 2019).

2.6.2.4 Evaluasi

Evaluasi adalah proses dalam menentukan nilai fitness pada setiap individu. Nilai fitness digunakan untuk menentukan jarak terpendek. Evaluasi digunakan untuk menghitung kebugaran (*fitness*) setiap *chromosome*. Semakin besar *fitness* maka semakin baik *chromosome* tersebut untuk dijadikan calon solusi. Karena sebuah *chromosome* selalu memiliki nilai *fitness* dan beberapa properti lain, maka dalam pembahasan berikutnya seringkali digunakan istilah 'individu'. Hal ini bisa

dianalogikan dengan seorang manusia sebagai individu. Dia memiliki tubuh beserta susunan gen pembentuknya (*chromosome*), Nama, umur, alamat dan properti lainnya (Mahmudy, 2015).

Sebagai ukuran performasinya, suatu individu atau kromosom akan melalui proses evaluasi berdasarkan fungsi tertentu. Agar ditemukan nilai kecocokan atau nilai optimalitasnya, maka digunakan fungsi yang disebut fungsi *fitness* (*fitness function*). Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Algoritme Genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi. Permasalahan TSP bertujuan meminimalkan jarak, maka nilai *fitness* adalah inversi dari total jarak dari jalur yang didapatkan (Sinaga, 2019).

2.6.2.5 Seleksi

Seleksi merupakan tahap paling akhir untuk menyeleksi individu dari semua solusi untuk menghasilkan nilai *fitness* yang diinginkan (Raditya & Dewi, 2018). Ada beberapa macam metode seleksi yang dapat digunakan adalah *roulette-wheel* dimana pada masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara professional sesuai dengan nilai *fitness*-nya. Sedangkan seleksi rangking dimulai dengan mengurutkan kromosom di dalam populasi berdasarkan *fitness*-nya kemudian memberi nilai *fitness* berdasarkan urutannya (Suwirmayanti, dkk., 2016). Pada penelitian ini, metode seleksi yang digunakan yaitu metode seleksi *roulette-wheel*.

Seleksi merupakan proses pemilihan individu terbaik untuk menjadi calon orang tua yang akan dilakukan proses selanjutnya. Proses pemilihan tersebut biasanya dipilih berdasarkan probabilitas dari individu yang terbaik dalam populasi. Individu yang terbaik ditentukan berdasarkan nilai *fitness* masing-masing dari tiap-tiap individu (Sinaga, 2019).

BAB 3 METODOLOGI PENELITIAN

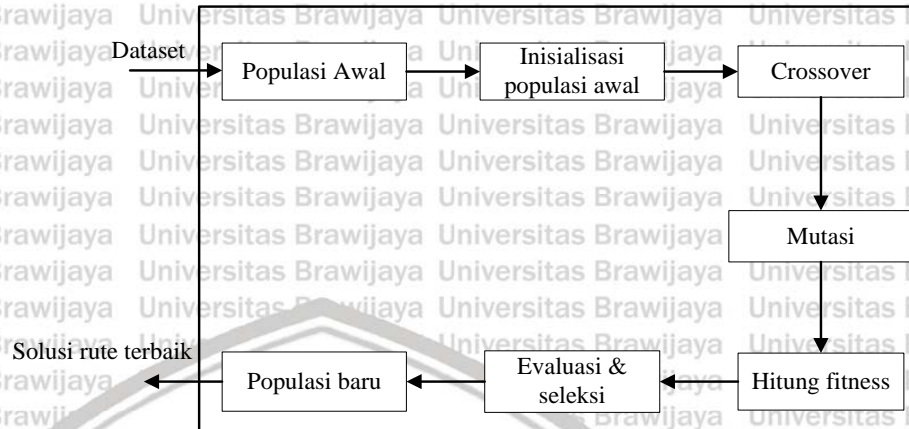
3.1 Tipe Penelitian

Tipe penelitian deskriptif merupakan sebuah penelitian yang dibuat dengan tujuan agar memberi gambaran atau deskripsi mengenai suatu situasi secara objektif.

Tipe penelitian ini dipakai untuk menjawab atau memecahkan sebuah permasalahan yang dihadapi di dalam situasi saat ini. Metode penelitian kuantitatif merupakan proses penelitian yang dapat menyelesaikan permasalahan yang berkaitan dengan data meliputi angka dan program statistik (Dr. Wahidmurni, 2017). Pendekatan kuantitatif merupakan salah satu upaya pencarian ilmiah (*scientific inquiry*) yang didasari oleh filsafat positivisme logikal (*logical positivism*) yang beroperasi dengan aturan–aturan yang ketat mengenai logika, kebenaran, hukum–hukum, dan prediksi. Fokus penelitian kuantitatif diidentifikasi sebagai proses kerja yang berlangsung secara ringkas, terbatas dan memilah–milah permasalahan menjadi bagian yang dapat diukur atau dinyatakan dalam angka–angka. (Surya Dharma, 2008)

3.2 Strategi Penelitian

Pada tahap strategi penelitian yang diperlukan adalah strategi penelitian berupa eksperimen. Metode Eksperimen digunakan untuk menjawab suatu masalah atau menguji suatu hipotesis. Suatu eksperimen berhasil apabila variabel yang dimanipulasi dinyatakan secara jelas dalam suatu hipotesis dan juga kondisi-kondisi yang dikontrol sudah tepat. Untuk hasilnya, maka setiap eksperimen harus di rancang terlebih dahulu baru kemudian di uji coba.



Gambar 3.1 Diagram Blok Optimasi Pengiriman Barang Dengan Algoritme Genetika

Dari diagram blok pada Gambar 3.1 dapat dijelaskan setiap langkah dari Algoritme Genetika. Data dalam penelitian akan dilakukan pemilihan populasi awal dari *dataset* yang didapatkan dari kantor JNE. Kemudian melakukan inisialisasi data awal hingga sesuai dengan gen dan kromosom yang telah dibentuk. Selanjutnya melakukan proses *crossover* dengan menentukan dua induk dari populasi secara tidak berurutan agar memperoleh nilai *offspring*. Setelah itu, dilakukan proses mutasi dengan menukarkan nilai gen secara acak sehingga menghasilkan *offspring*. Lalu menghitung nilai *fitness* untuk menantukan jarak terpendek. Kemudian proses evaluasi yang merupakan proses dalam menentukan nilai *fitness* pada setiap individu. Langkah proses seleksi yang dilakukan untuk memilih individu dari semua solusi agar menghasilkan nilai *fitness* yang diinginkan. Kemudian didapatkan populasi baru yang merupakan solusi rute terbaik.

3.3 Partisipan Penelitian

Partisipan penelitian yang terlibat adalah karyawan JNE yang berada di Kantor. Alasan untuk memilih karyawan JNE yaitu karena informasi yang di dapat lebih akurat. Karyawan yang membantu adalah staf karyawan bagian *entry* data yang dapat membantu memberi Data alamat pengirim yaitu alamat untuk mengirimkan barang, nama pengirim yaitu nama dari penerima, nama kurir yaitu nama pegawai dari JNE. Tentu hal ini dibawah pengawasan manajer staf bagian *entry* data.

3.4 Lokasi Penelitian

Dalam penelitian ini data yang digunakan diambil dari Kantor JNE yang berada di Jl. Sunan Drajat No.332 E, Kaloharjo, Sidoharjo, Kec. Lamongan, Kabupaten Lamongan. Pemilihan lokasi ini berdasarkan dengan kesediaan kantor JNE tersebut memberikan data. Data dapat diperoleh melalui persyaratan dan persetujuan dari pemimpin Kantor tersebut sesuai dengan peraturan yang berlaku. Peraturan dalam hal ini adalah bentuk perijinan dari pihak kampus dan persetujuan dari pihak Kantor JNE. Selain itu data yang diberikan juga terbatas pada data yang memang diijinkan dari pihak Kantor JNE. Data tersebut berupa alamat pengirim yaitu alamat untuk mengirimkan barang, nama pengirim yaitu nama dari penerima, nama kurir yaitu nama pegawai dari JNE.

3.5 Teknik Pengumpulan Data

Pada tahap pengumpulan data ini data yang digunakan adalah data primer. Data primer yaitu data yang diperoleh dari sumber asli atau pertama. Data harus dicari melalui narasumber, yaitu orang yang dijadikan sebagai sarana untuk mendapatkan informasi atau data. Data yang digunakan pada penelitian ini terdapat beberapa jenis yaitu alamat pengirim yaitu alamat untuk mengirimkan barang, Nama Pengirim yaitu nama dari penerima, Nama Kurir yaitu nama pegawai dari JNE yang ditugaskan untuk mengirimkan barang. Dari Data Nama Pengirim maka didapatkan alamat pengiriman barang sehingga ditemukan titik alamat pengiriman barang yang kemudian digunakan untuk menemukan jarak antar penerima barang. Dimana jarak antar penerima barang digunakan untuk menghitung nilai *fitness*.

3.6 Data Penelitian

Data yang diolah pada penelitian ini adalah Nama Pengirim dan Data Alamat dengan alamat yang berbeda-beda. Dimana data didapat dari alamat pengirim yaitu alamat untuk mengirimkan barang, Nama Pengirim yaitu nama dari penerima, Nama Kurir yaitu nama pegawai dari JNE. Dari data alamat pengiriman maka dibuatlah titik alamat pengiriman barang sehingga memudahkan untuk mengetahui jarak antara satu alamat dengan alamat yang lain. Jarak ini dapat digunakan untuk mendapatkan nilai *fitness*.

3.7 Teknik Analisis Data

Pada tahap analisis data ini dilakukan pegujian data dengan menggunakan metode Algoritme Genetika. Pada metode Algoritma Genetika terdapat beberapa tahap yaitu representasi kromosom, inialisasi kromosom, *crossover*, mutasi, evaluasi, seleksi. Analisis data yang dipakai yaitu Analisis Deskriptif. Analisis Deskriptif dilakukan dengan cara mendeskripsikan masing-masing variabel penelitian berdasar data yang diperoleh.

3.8 Implementasi Algoritme

Pada implementasi algoritme adalah proses untuk melakukan manualisasi supaya mempermudah dalam setiap langkah perhitungan dalam metode Algoritme Genetika. Kemudian menentukan alur perhitungan Algoritme Genetika. Dimana implementasi algoritme akan dibuat berdasarkan dengan diagram blok yang telah dibuat. Implementasi insialisasi populasi awal yang berisi gen dan kromosom yang sesuai dengan data. Implementasi *crossover* yang dilakukan untuk mendapatkan kromosom yang lebih baik. Implementasi mutasi yang dilakukan dengan menggeser nilai gen terhadap gen yang sudah terpilih. Implementasi Evaluasi & seleksi yang digunakan untuk mendapatkan nilai *fitness* terbaik sebagai solusi rute terpendek yang akan ditemukan.

3.9 Peralatan Pendukung

Dalam penelitian ini, alat pendukung yang digunakan adalah yang dapat menunjang proses penelitian dengan baik. Dalam hal ini termasuk menggunakan perangkat lunak dan perangkat keras.

1) Perangkat Keras

Perangkat keras atau *hardware* yang digunakan sebagai pendukung dalam proses penelitian ini sebagai berikut:

- Laptop Asus X442UQ

- Layar Monitor 14"

- Memori RAM 8074MB

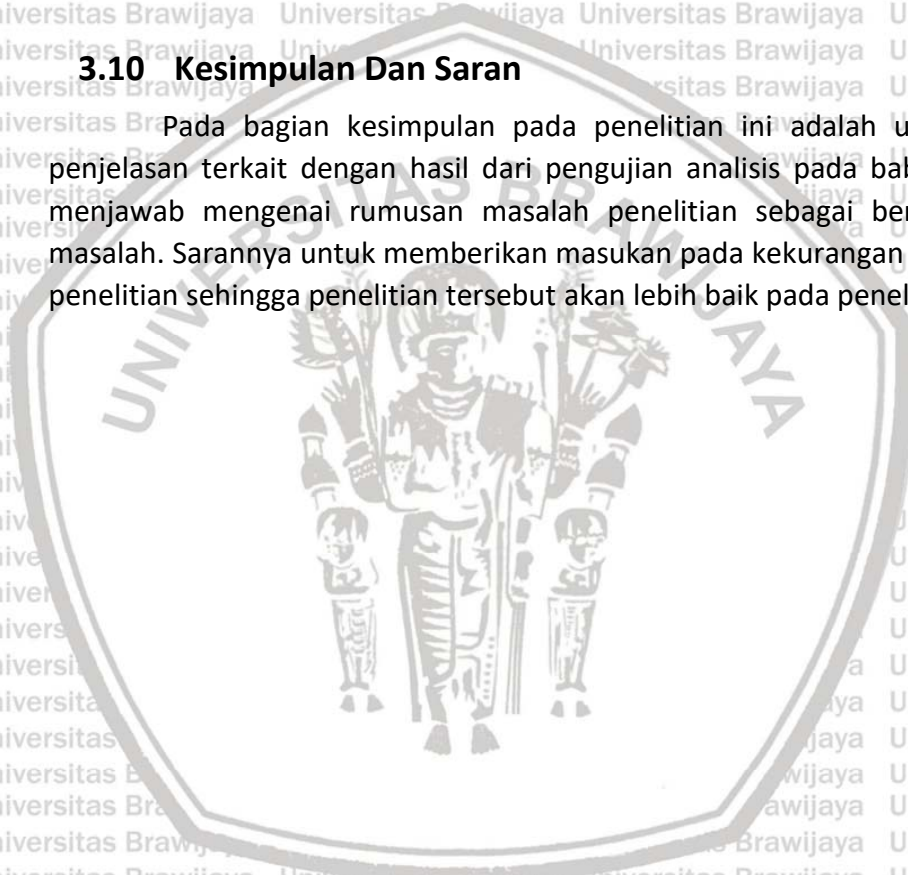
2) Perangkat Lunak

Perangkat lunak atau *software* yang digunakan sebagai pendukung dalam proses penelitian ini sebagai berikut:

- Sistem operasi Windows 10
- NetBeans IDE 8.2
- MySQL

3.10 Kesimpulan Dan Saran

Pada bagian kesimpulan pada penelitian ini adalah untuk memberikan penjelasan terkait dengan hasil dari pengujian analisis pada bab sebelumnya dan menjawab mengenai rumusan masalah penelitian sebagai bentuk penyelesaian masalah. Sarannya untuk memberikan masukan pada kekurangan yang didapat pada penelitian sehingga penelitian tersebut akan lebih baik pada penelitian selanjutnya.



BAB 4 PERANCANGAN

Pada bab perancangan ini menjelaskan tentang proses analisis kebutuhan dan susunan kerja dalam perancangan optimasi pengiriman barang menggunakan Algoritma Genetika, mulai dari perancangan model sampai dengan antar muka sistem.

4.1 Deskripsi Masalah

Permasalahan pada penelitian ini merupakan proses pengiriman barang ke penerima yang dilaksanakan oleh banyak kurir dan dibutuhkan penyelesaian dengan menggunakan *Multiple Travelling Salesman Problem (M-TSP)*. Penelitian ini menggunakan tiga kurir. Dalam menyelesaikan permasalahan ini membutuhkan data jarak antar penerima barang supaya ke tiga kurir dapat mengunjunginya. Pada tabel jarak terdapat beberapa baris dan kolom yang berisi jumlah penerima yang Akan dikunjungi oleh kurir untuk mengirimkan barang. Di dalam tabel jarak terdapat nilai 0 yang dimaksud adalah jarak dari Kantor JNE ke tempat tujuan atau ke penerima barang. Data jarak dari titik awal yaitu Kantor JNE menuju ke tempat tujuan dapat dilihat pada Lampiran B yang berada di akhir laporan ini.

Penyelesaian masalah pada penelitian pengiriman barang ini yaitu bagaimana setiap kurir supaya bisa menempuh rute dengan total jarak minimum agar kurir dapat mengirimkan barang ke penerima. Adanya beberapa barang ini, dapat dicontohkan ada tiga kurir yang mengirimkan barang ke penerima. Pada tiap kurir akan di berikan solusi dengan mengurutkan daftar penerima barang yang harus didatangi, tetapi bisa juga kurir satu dengan kurir lainnya membawa barang yang jumlahnya berbeda-beda. Data jarak dari titik awal yaitu Kantor JNE menuju ke tempat tujuan dapat dilihat pada Tabel 4.1 dan hasil secara detail terdapat pada Lampiran B.

Tabel 4.1 Data Jarak Tiap Penerima

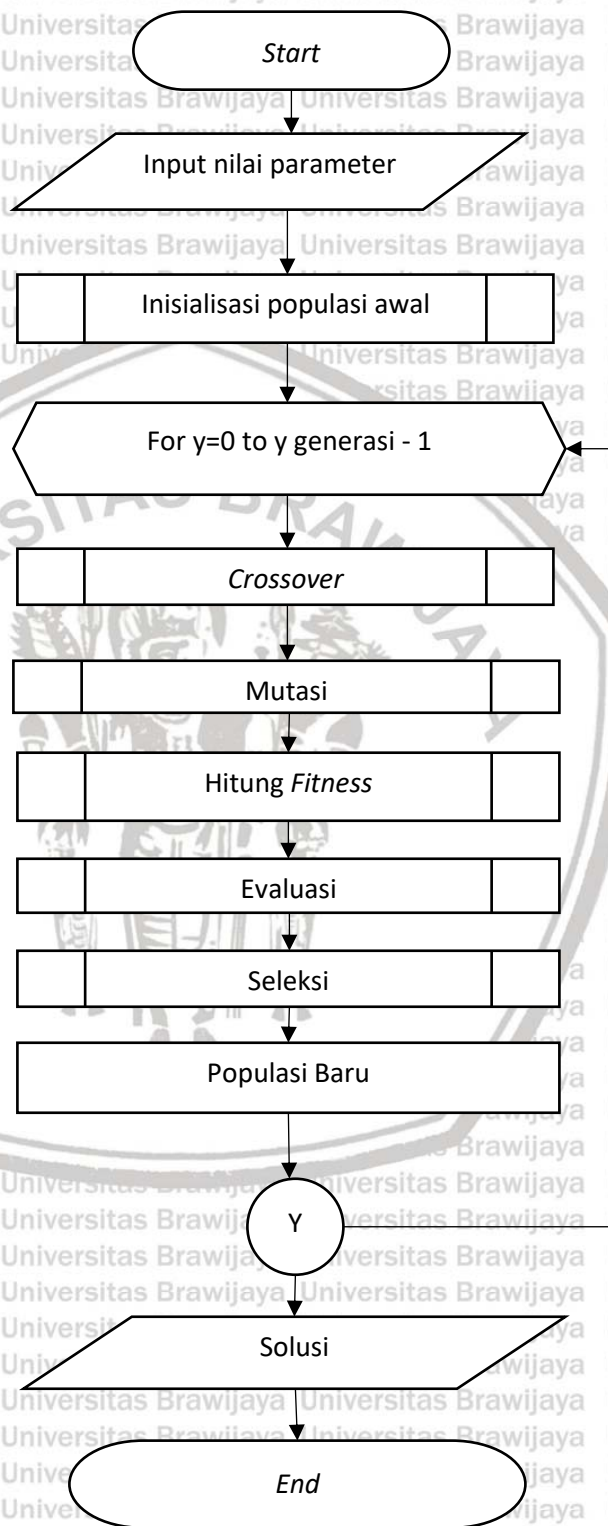
y	0	1	2	3	4	5	...	11	12	13	14	15
0	0	3,5	2,2	2,7	6,2	2,0	...	3,0	2,5	1,8	1,3	2,3
1	3,5	0	4,4	1,0	6,9	2,1	...	3,8	3,1	4,0	2,4	1,4
2	2,2	4,4	0	1,5	5,6	3,0	...	5,2	2,9	2,2	2,7	1,9
3	2,7	1,0	1,5	0	7,2	1,8	...	1,1	4,6	3,6	3,0	4,7
4	6,2	6,9	5,6	7,2	0	6,7	...	6,0	7,8	5,4	6,7	4,4

5	2,0	2,1	3,0	1,8	6,7	0	...	0,6	1,5	0,9	8,2	7,4
6	1,7	2,9	3,1	3,9	1,2	4,8	...	2,7	0,4	6,7	12,3	8,3
7	1,1	2,4	8,2	2,2	11,0	9,3	...	3,2	13,9	1,3	10,8	12,6
8	4,5	3,4	4,4	6,3	2,7	5,6	...	1,1	11,6	13,2	3,1	6,3
9	3,3	4,5	3,9	4,0	5,2	9,7	...	12,0	5,0	2,3	13,3	10,1
10	5,1	3,7	2,8	3,2	3,9	2,1	...	10,0	14,9	11,9	3,3	1,3
11	3,0	3,8	5,2	1,1	6,0	0,6	...	0	3,2	3,8	15,0	11,8
12	2,5	3,1	2,9	4,6	7,8	1,5	...	3,2	0	8,3	1,2	0,7
13	1,8	4,0	2,2	3,6	5,4	0,9	...	3,8	8,3	0	6,0	13,5
14	1,3	2,4	2,7	3,0	6,7	8,2	...	15,0	1,2	6	0	4,2
15	2,3	1,4	1,9	4,7	4,4	7,4	...	11,8	0,7	13,5	4,2	0

Pada Tabel 4.1 menjelaskan tentang data jarak dari alamat penerima barang dari JNE yang sedang dilakukan penelitian. Kemudian terdapat jarak dari alamat penerima barang satu ke alamat penerima barang yang lain. Sehingga data jarak dari 15 alamat penerima barang dapat saling berhubungan satu dengan yang lainnya dan dapat memudahkan kurir dalam mengirim barang pada alamat yang akan dituju.

4.2 Siklus Algoritme Genetika

Pada permasalahan pengiriman barang ini dapat diselesaikan dengan menggunakan Algoritme Genetika yang bisa memberikan solusi alternatif paling optimal. Langkah-langkah metode Algoritme Genetika dalam menyelesaikan masalah yang di tampilkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Algoritme Genetika

Penelitian tentang optimasi rute pengiriman barang JNE ini menggunakan metode Algoritme Genetika. Karena metode Algoritma Genetika ini dapat dipakai untuk menyelesaikan masalah optimasi dengan model matematika yang kompleks (Mahmudy, 2015).

Diagram pada Gambar 4.1 terdapat langkah-langkah Algoritme Genetika yang bisa digunakan untuk menyelesaikan pengiriman barang, sebagai berikut:

1) Inisialisasi parameter

Inisialisasi parameter pada masalah ini ada dua yaitu yang pertama parameter pengiriman dan yang ke dua yaitu parameter Algoritme Genetika. Berikut adalah parameter pengiriman yang digunakan:

- a. Alamat pengirim yaitu alamat untuk mengirimkan barang.
- b. Nama Pengirim yaitu Nama dari penerima.
- c. Nama Kurir yaitu Nama pegawai dari JNE yang ditugaskan untuk mengirimkan barang.

2) Representasi kromosom dengan menginputkan nilai parameter Algoritma Genetika sebagai berikut:

- a. Ukuran populasi (*popSize*) adalah jumlah individu yang dimasukkan pada setiap generasi.
- b. *Crossover Rate* (Cr) adalah rasio perbandingan banyaknya *offspring* yang dibuat pada setiap generasi dengan banyaknya *population size* (Laksono, dkk., 2016).
- c. *Mutation Rate* (Mr) adalah representasi terjadinya kromosom baru sebagai akibat mutasi dari keseluruhan *population size* (Laksono, dkk., 2016).
- d. Generasi merupakan banyaknya jumlah generasi yang diproses pada proses Algoritme Genetika.

3) Inisialisasi populasi awal diproses dengan memberikan nilai awal gen-gen dengan nilai acak berdasarkan limitasi yang telah dideksipkan sebelumnya.

4) Reproduksi

- a. *Crossover* merupakan proses menyilangan antar *string* yang sudah diperoleh dari reproduksi sebelumnya.
- b. *Mutation* merupakan proses untuk mengubah nilai dari satu atau beberapa gen pada kromosom

5) Evaluasi dan Seleksi

Evaluasi adalah proses dalam menentukan nilai *fitness* pada setiap individu, sedangkan Seleksi adalah tahap untuk memilih individu dari semua solusi untuk menghasilkan *fitness* yang diinginkan.

4.3 Siklus Penyelesaian Masalah Menggunakan Algoritme Genetika

Dalam menyelesaikan permasalahan pada penelitian ini adalah terkait pengoptimasian pengiriman barang JNE di Lamongan dengan menggunakan Algoritme Genetika. Penelitian ini bertujuan untuk mendapatkan rekomendasi pengiriman barang yang paling efektif dan cepat. Pengiriman barang melibatkan beberapa kurir untuk proses pengiriman barang ke penerima, sehingga permasalahan ini bisa diselesaikan dengan menggunakan M-TSP dengan metode yang digunakan yaitu metode Algoritme Genetika.

4.3.1 Representasi Kromosom

Pada penelitian ini, representasi kromosom ini memiliki panjang *string* kromosom sebanyak 18 terdiri dari dua segmen. Segmen pertama yaitu 15 urutan pengiriman barang dan segmen ke dua yaitu jumlah barang yang harus dibawa oleh setiap kurir. Jumlah barang yang dibawa pada setiap kurir memiliki jumlah yang berbeda-beda tetapi jumlah barang yang dibawa semua kurir sama dengan jumlah keseluruhan barang yang harus dikunjungi. Representasi kromosom ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Representasi Kromosom

p	Kromosom																	
	Segmen 1															Segmen 2		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6	7	4	4

Representasi kromosom pada Tabel 4.2 terdapat 2 segmen, yaitu:

1. Segmen 1

Merupakan urutan untuk pengiriman barang ke penerima. Pada segmen 1 terdapat 15 angka, mulai dari angka 1 hingga angka 15. Yang mana pada segmen 1

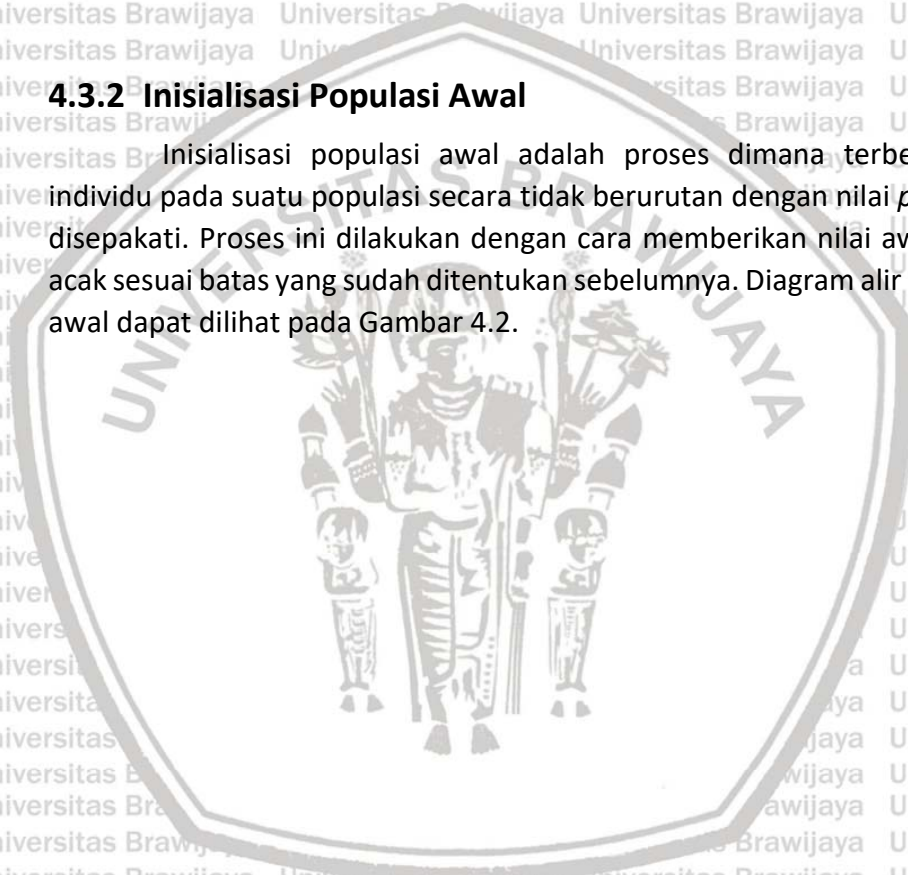
ini merupakan antrian pengiriman barang secara acak pada setiap penerima barang.

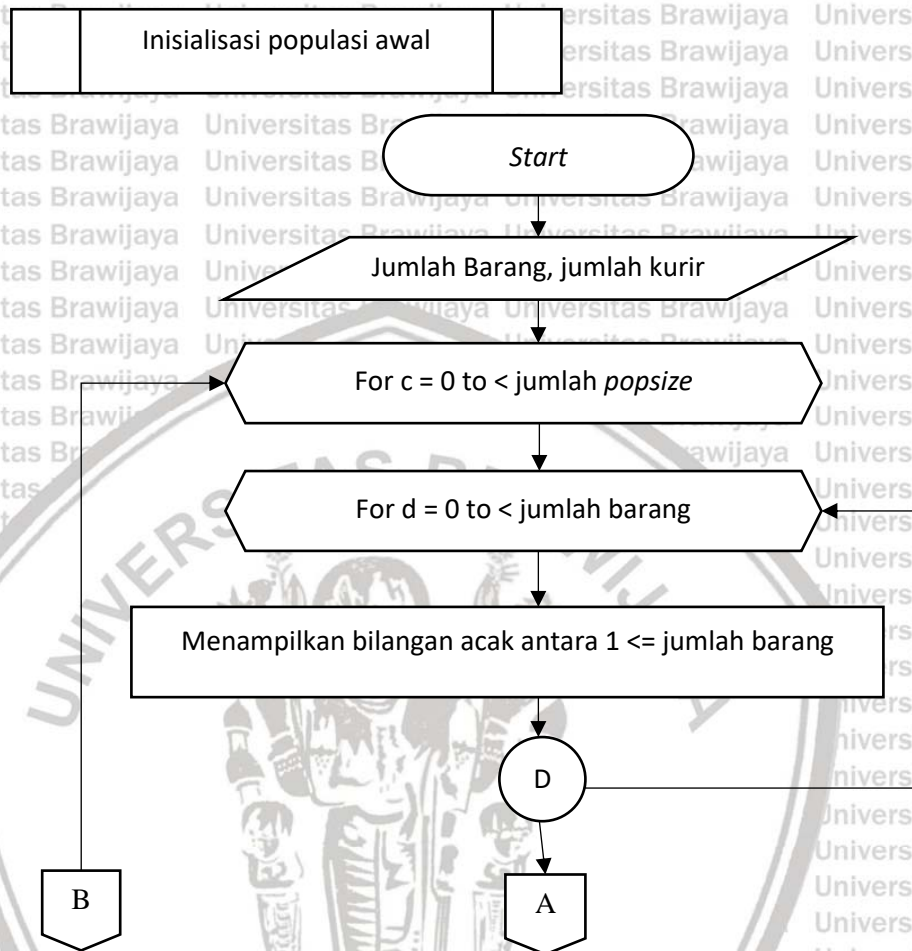
2. Segmen 2

Merupakan jumlah barang yang dibawa oleh setiap kurir. Pada segmen 2 terdapat 3 angka lanjutan dari segmen 1 yaitu 16, 17, dan 18. Dimana pada segmen 2 ini merupakan jumlah barang yang dikirim pada setiap kurir.

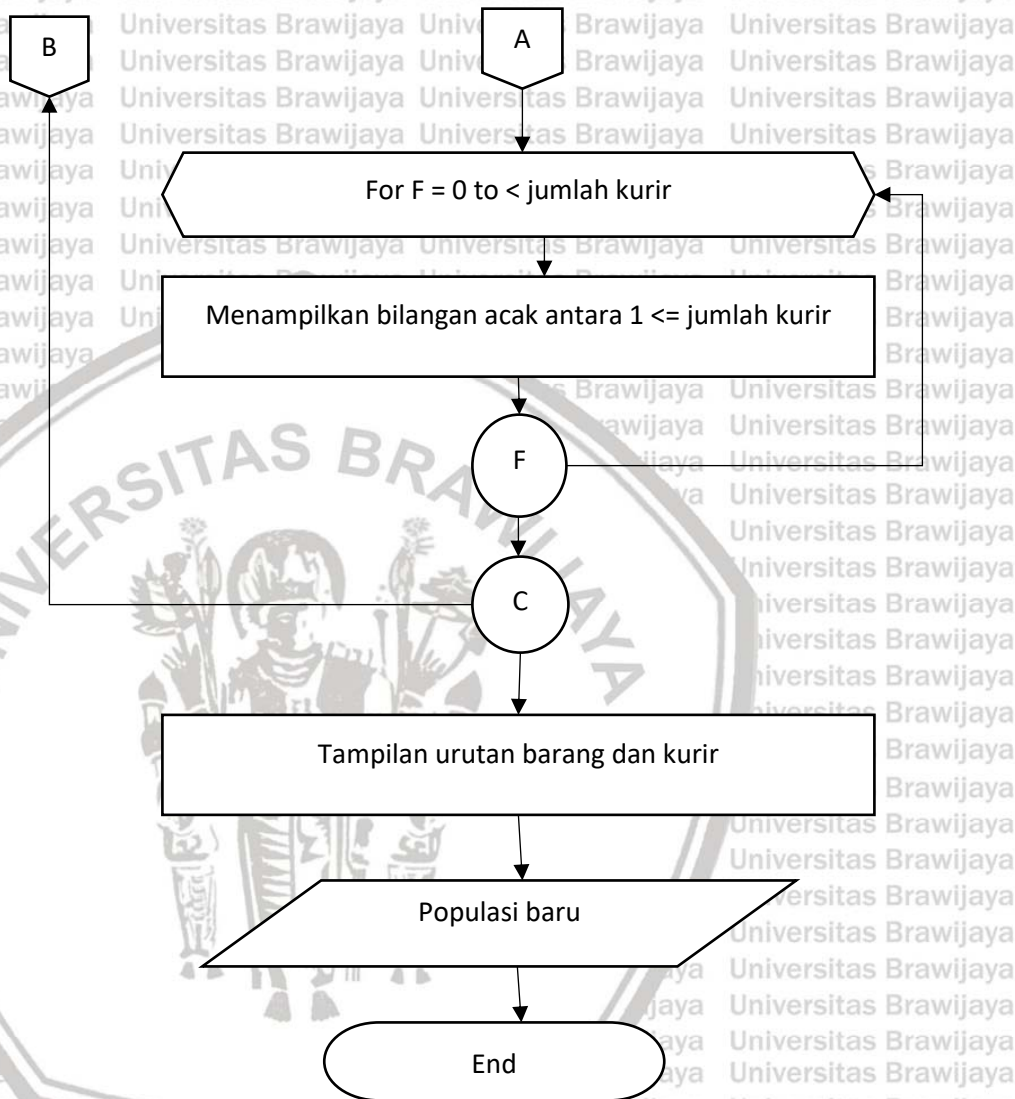
4.3.2 Inisialisasi Populasi Awal

Inisialisasi populasi awal adalah proses dimana terbentuknya individu-individu pada suatu populasi secara tidak berurutan dengan nilai *popSize* yang sudah disepakati. Proses ini dilakukan dengan cara memberikan nilai awal gen-gen secara acak sesuai batas yang sudah ditentukan sebelumnya. Diagram alir inisialisasi populasi awal dapat dilihat pada Gambar 4.2.





Gambar 4.2 Diagram Inisialisasi Populasi Awal



Gambar 4.2 Diagram Inisialisasi Populasi Awal (Lanjutan)

Dari diagram pada Gambar 4.2 dapat dijelaskan pada setiap langkah dari proses inisialisasi populasi awal yaitu sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah data jumlah barang dan data jumlah kurir.
2. Melakukan perulangan variabel $c=0$ hingga kurang dari ((panjang *dataset*)-*popsiz*e)
3. Melakukan perulangan variabel $d=0$ hingga kurang dari ((panjang *dataset*)-jumlah barang)

4. Menampilkan bilangan acak antara 1 hingga kurang dari sama dengan jumlah barang
 5. *Loop* pada variabel d (langkah ke 3) hingga data barang habis
 6. Melakukan perulangan variabel f=0 hingga kurang dari ((panjang *dataset*)-jumlah kurir)
 7. Menampilkan bil. Acak antara 1 hingga kurang dari sama dengan jumlah kurir
 8. *Loop* pada variabel f (langkah 6) hingga data kurir habis
 9. *Loop* pada variabel c (langkah 2) hingga semua data habis
 10. Tampilan urutan barang dan kurir
 11. Didapatkan *output* populasi baru
- Inisialisasi populasi awal pada penelitian ini yaitu *popSize* sebanyak 5. Ada 5 data yang digunakan yang dapat dilihat pada Tabel 4.3.

Tabel 4.3 Inisialisasi Populasi Awal

P	Kromosom																	
	Segmen 1															Segmen 2		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
p1	3	11	7	1	13	10	5	14	8	12	4	2	13	9	6	7	4	4
p2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4	3	8	4
p3	7	4	6	11	13	8	2	5	10	12	1	3	15	9	14	6	4	5
p4	2	9	12	4	15	6	10	3	14	8	5	11	7	1	13	5	5	8
p5	12	4	14	11	8	13	7	10	1	15	6	9	3	5	2	4	7	6

Pada Tabel 4.3 menjelaskan populasi awal yang digunakan untuk sebagai perhitungan awal. Dimana data yang digunakan merupakan data pengirim barang sebanyak 15 data dan dengan menggunakan 3 kurir barang. Kemudian setiap urutan alamat diacak sebanyak *popsiz* (*popsiz*=5). Sehingga terdapat 5 urutan alamat data pengirim yang ada dengan kurir yang berbeda pula. Setelah populasi sebanyak *popsiz* didapatkan maka dilakukan proses Algoritma Genetika hingga mendapatkan hasil *fitness* akhir pada masing – masing urutan data alamat yang sebanyak *popsiz* tadi. Dari hasil *fitness* yang didapatkan maka akan dipilih mana jalur yang memiliki nilai *fitness* tertinggi. Jalur dengan nilai *fitness* tertinggi yang akan dipilih sebagai alternatif terbaik bagi kurir untuk mengirim paket pada alamat yang tertera pada urutan nilai *fitness* terbaik tersebut.

4.3.3 Reproduksi

Tahap selanjutnya adalah proses reproduksi. Proses reproduksi adalah proses untuk menghasilkan keturunan baru dengan cara mewariskan sifat-sifat yang sama dari kromosom induk (Hannawati, dkk., 2002).

4.3.3.1 Crossover

Crossover adalah proses persilangan yang melibatkan dua induk untuk mendapatkan keturunan baru (Sofwan, dkk., 2008). Pada proses ini memilih dua individu pada populasi secara acak kemudian mencari titik potong menggunakan Metode *One Cut Point*. *One Cut Point* yaitu proses pemilihan secara acak pada satu posisi pada kromosom induk kemudian saling bertukar gen (Lubis & Ginting, 2016). Alur diagram *crossover* dapat dilihat pada Gambar 4.3, sedangkan untuk diagram alir proses *crossover* dapat dilihat pada Gambar 4.4.

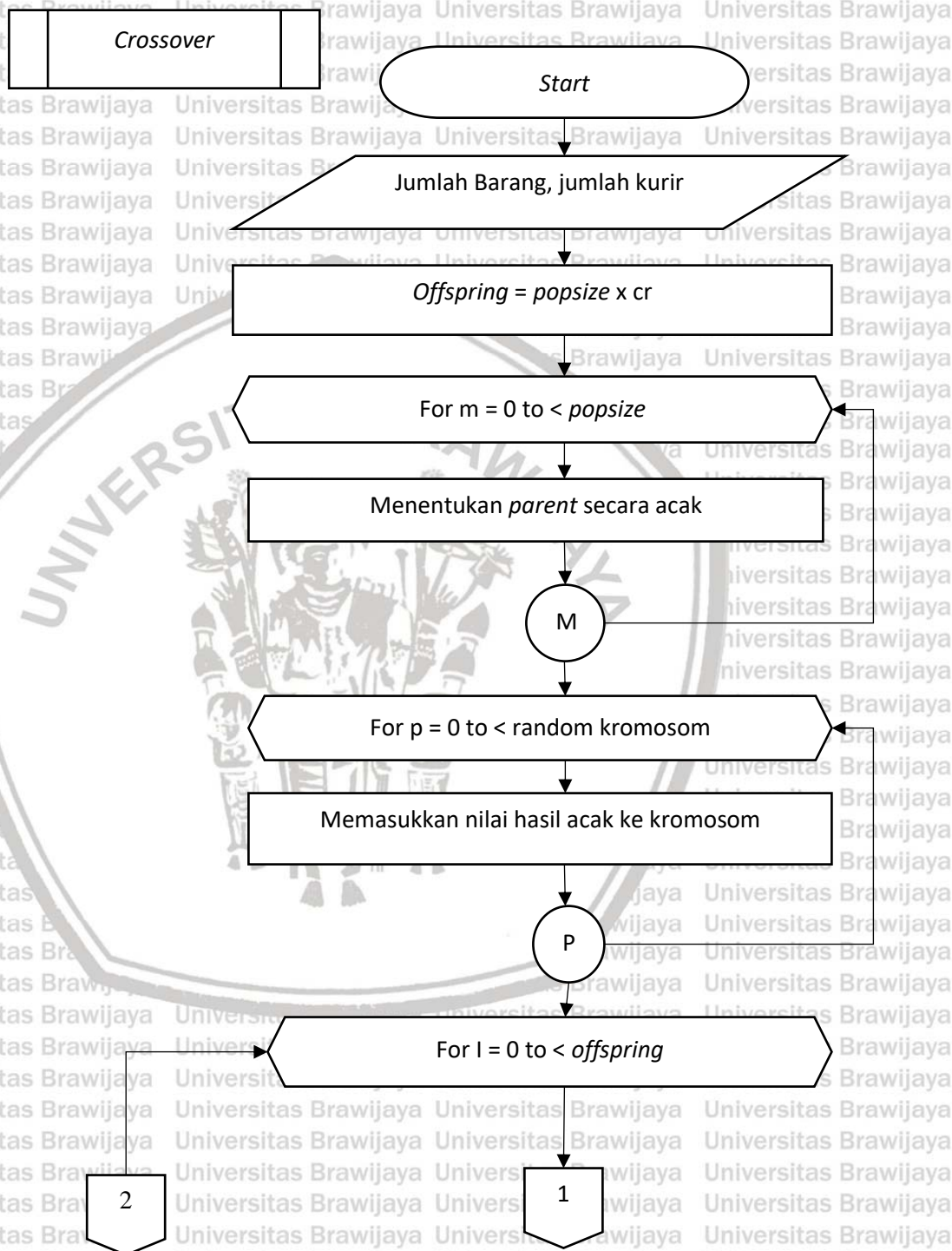
$$Offspring = cr \times popSize \quad (4.1)$$

$$Offspring = 0,4 \times 5 = 2 \quad (4.2)$$

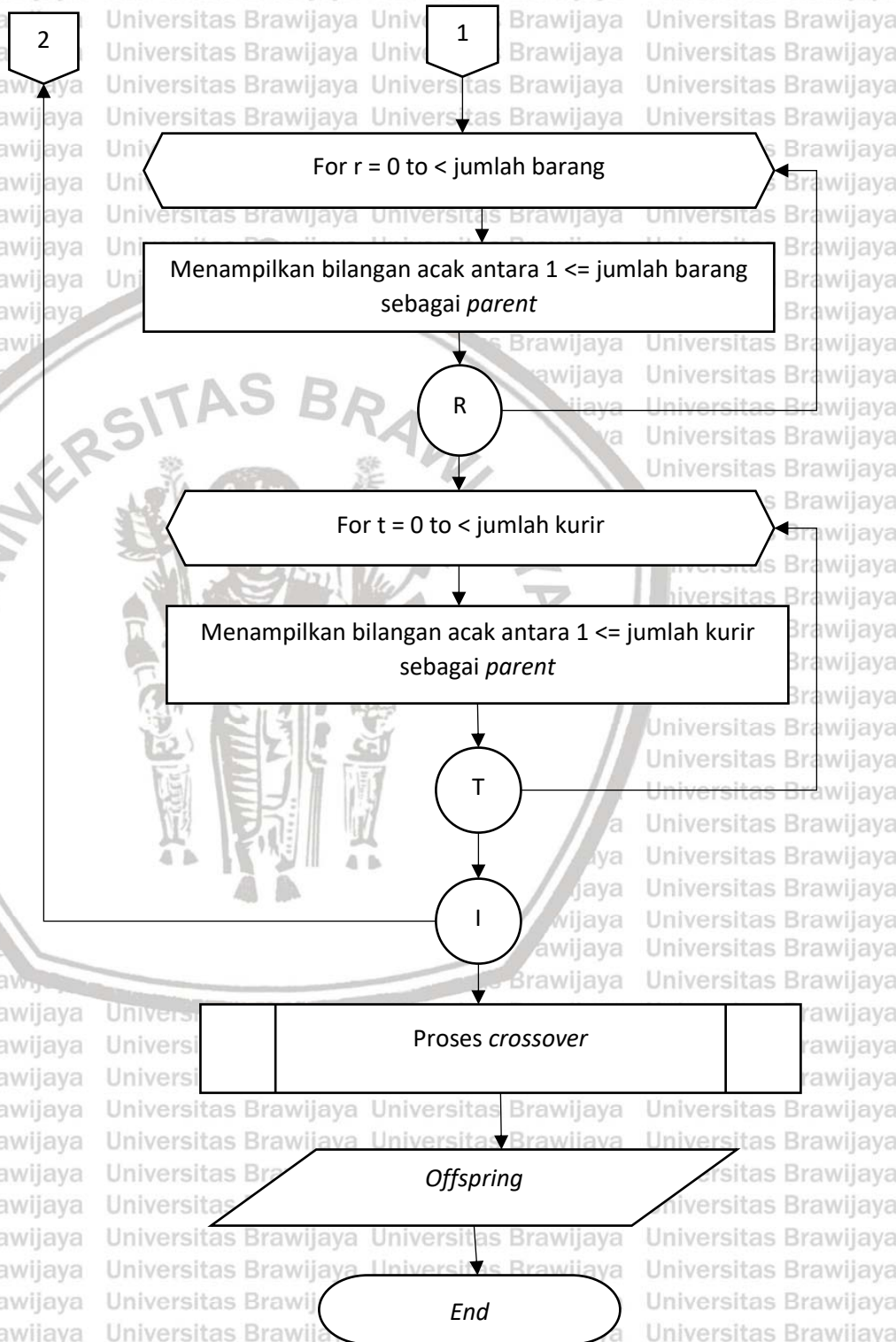
Pada Gambar 4.3 dapat dideskripsikan setiap proses dari *Crossover* sebagai berikut:

- 1) Menambahkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah data jumlah barang dan data jumlah kurir.
- 2) Hitung $offspring = popsize \times cr$
- 3) Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-*popsize*)
- 4) Menentukan *parent* secara acak
- 5) *Loop* pada variabel m (langkah 3) hingga mencapai batas *popsize*
- 6) Melakukan perulangan variabel $p=0$ hingga kurang dari ((panjang *dataset*)-*random* kromosom)
- 7) Memasukkan nilai hasil acak ke kromosom
- 8) *Loop* pada variabel p (langkah 6) hingga data *random* kromosom habis

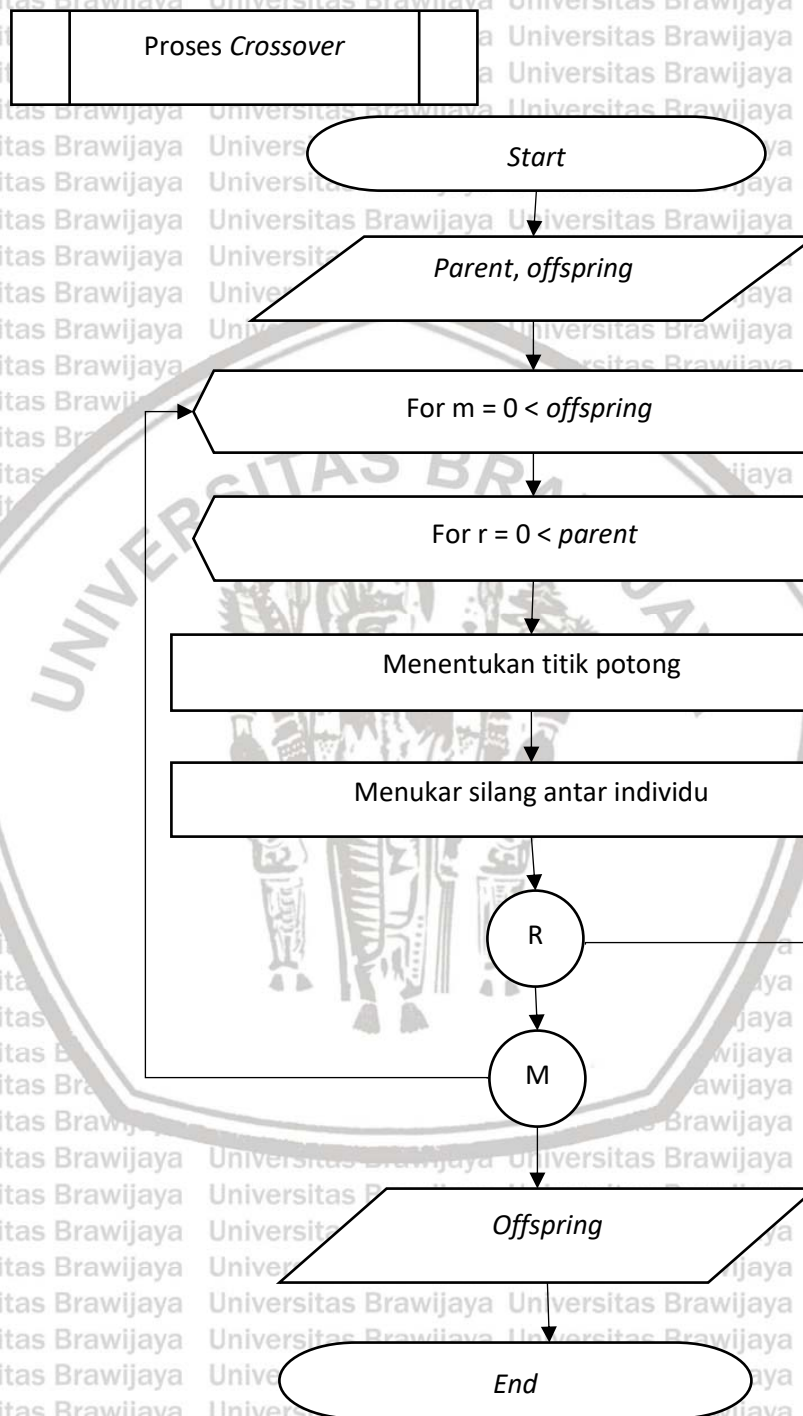
- 9) Melakukan perulangan variabel $l=0$ hingga kurang dari ((panjang *dataset*)-*offspring*)
- 10) Melakukan perulangan variabel $r=0$ hingga kurang dari ((panjang *dataset*)-jumlah barang)
- 11) Menampilkan bilangan acak antara 1 hingga kurang dari sama dengan jumlah barang sebagai *parent*
- 12) *Loop* pada variabel r (langkah 10) hingga data barang habis
- 13) Melakukan perulangan variabel $t=0$ hingga kurang dari ((panjang *dataset*)-jumlah kurir)
- 14) Menampilkan bilangan acak antara 1 hingga kurang dari sama dengan jumlah kurir sebagai *parent*
- 15) *Loop* pada variabel t (langkah 13) hingga data barang habis
- 16) *Loop* pada variabel l (langkah 9) hingga semua data habis
- 17) Memanggil proses *crossover*
- 18) Didapatkan output *offspring*



Gambar 4.3 Diagram Alir Crossover



Gambar 4.3 Diagram Alir Crossover (Lanjutan)



Gambar 4.4 Diagram Proses Crossover

Pada diagram di Gambar 4.4 dapat dideskripsikan setiap proses dari Crossover sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah *parent* dan *offspring*.
2. Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-*offspring*)
3. Melakukan perulangan variabel $r=0$ hingga kurang dari ((panjang *dataset*)-*parent*)
4. Menentukan titik potong
5. Menukar silang antar individu
6. Loop pada variabel r (langkah 2) hingga mencapai batas *offspring*
7. Loop pada variabel m (langkah 3) hingga data random *parent*
8. Didapatkan *output* berupa *offspring*

Proses *crossover* melakukan *cut point* pada titik ke-5. Proses *crossover* diawali dengan memilih titik *cut point* pada $p1$ dan $p2$, kemudian kedua induk tersebut di salin ke *child* ($c1$ & $c2$). Setelah itu, mulai dari titik *cut point* di cek pada $p1$ dan $p2$, kemudian sisa gen yang belum ada pada *child* dimasukkan. Perhitungan manualisasi *crossover* dapat dilihat pada Tabel 4.4 sebagai berikut:

Tabel 4.4 Proses Crossover

	Cut point ↓														
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6
P2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4
C1	3	11	7	1	13	10	5	2	9	15	6	14	12	8	4
C2	5	2	9	15	11	6	13	3	7	1	10	14	8	12	4

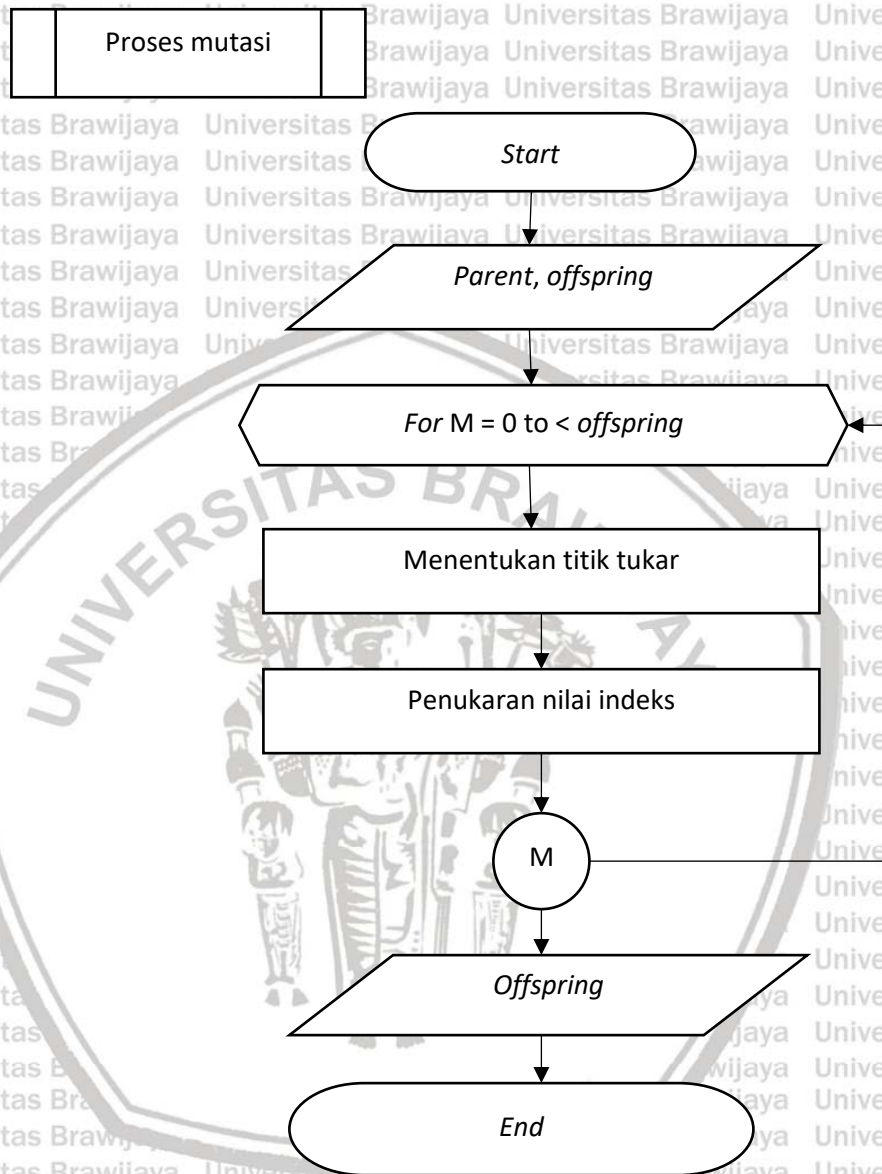
Pada Tabel 4.4 menjelaskan bahwa *parent* yang dipilih secara acak sebelumnya sama sekali tidak mengalami perubahan karena yang dilakukan proses *crossover* adalah *parent* yang dipilih. Untuk *parent* yang tidak terpilih yang dilakukan pertukaran data (*crossover*) atau juga dapat mengubah *parent* secara acak. Setelah itu data yang telah diacak akan dipindah pada tabel *child*.

4.3.3.2 Mutation

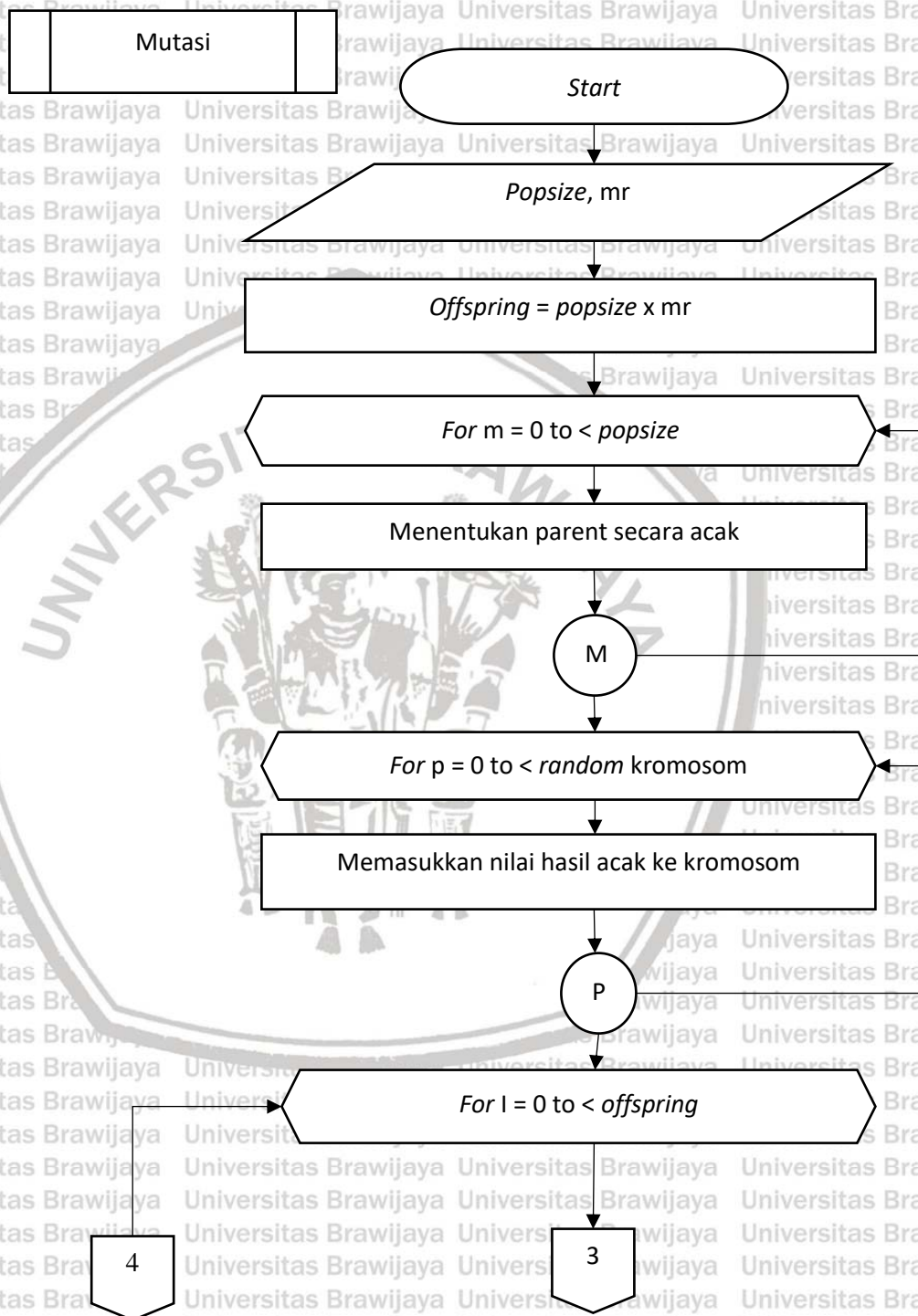
Mutasi adalah langkah untuk mentransformasi nilai dari gen-gen pada suatu kromosom. Mutasi dapat menghasilkan individu baru dengan cara mengubah satu atau lebih gen pada individu yang sama (Suwirmayanti, dkk., 2016). Proses mutasi dilakukan untuk menggantikan satu gen yang sudah terpilih secara acak dengan nilai gen yang diperoleh secara acak juga (Lubis & Ginting, 2016). Alur diagram proses mutasi dapat dilihat pada Gambar 4.5 sedangkan untuk diagram alir mutasi dapat dilihat pada Gambar 4.6.

Dari diagram pada Gambar 4.5 dapat dideksripsikan setiap alur dalam proses mutasi adalah sebagai berikut:

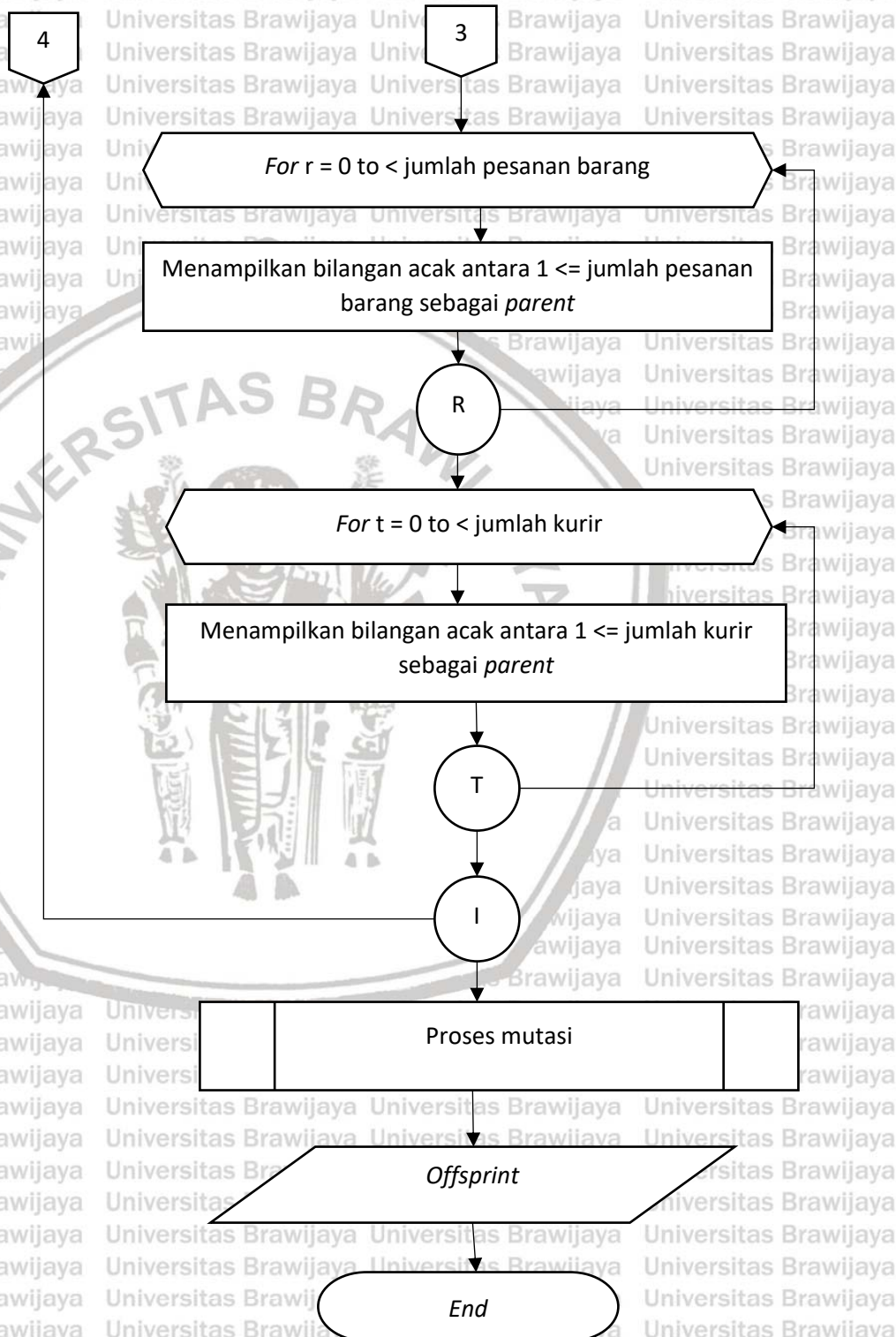
1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah *parent* dan *offspring*.
2. Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-*offspring*)
3. Menentukan titik tukar
4. Penukaran nilai indeks
5. *Loop* pada variabel m (langkah 3) hingga data *random parent*
6. Didapatkan *output* berupa *offspring*



Gambar 4.5 Diagram Proses Mutasi



Gambar 4.6 Diagram Alir Mutasi



Gambar 4.6 Diagram Alir Mutasi (Lanjutan)

Pada diagram di Gambar 4.6 dapat dideskripsikan setiap proses dari mutasi sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah *popsiz*, *mr*.
2. Hitung $offspring = popsize \times mr$
3. Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-*popsiz*)
4. Menentukan *parent* secara acak
5. *Loop* pada variabel m (langkah 3) hingga mencapai batas *popsiz*
6. Melakukan perulangan variabel $p=0$ hingga kurang dari ((panjang *dataset*)-*random* kromosom)
7. Memasukkan nilai hasil acak ke kromosom
8. *Loop* pada variabel p (langkah 6) hingga data *random* kromosom habis
9. Melakukan perulangan variabel $l=0$ hingga kurang dari ((panjang *dataset*)-*offspring*)
10. Melakukan perulangan variabel $r=0$ hingga kurang dari ((panjang *dataset*)-jumlah barang)
11. Menampilkan bilangan acak antara 1 hingga kurang dari sama dengan pesanan barang sebagai *parent*
12. *Loop* pada variabel r (langkah 10) hingga data barang habis
13. Melakukan perulangan variabel $t=0$ hingga kurang dari ((panjang *dataset*)-jumlah kurir)
14. Menampilkan bilangan acak antara 1 hingga kurang dari sama dengan jumlah kurir sebagai *parent*
15. *Loop* pada variabel t (langkah 13) hingga data barang habis
16. *Loop* pada variabel l (langkah 9) hingga semua data habis
17. Memanggil proses mutasi
18. Didapatkan *output offspring*

Langkah-langkah perhitungan manualisasi mutasi dapat dilihat pada Tabel 4.5 sebagai berikut:

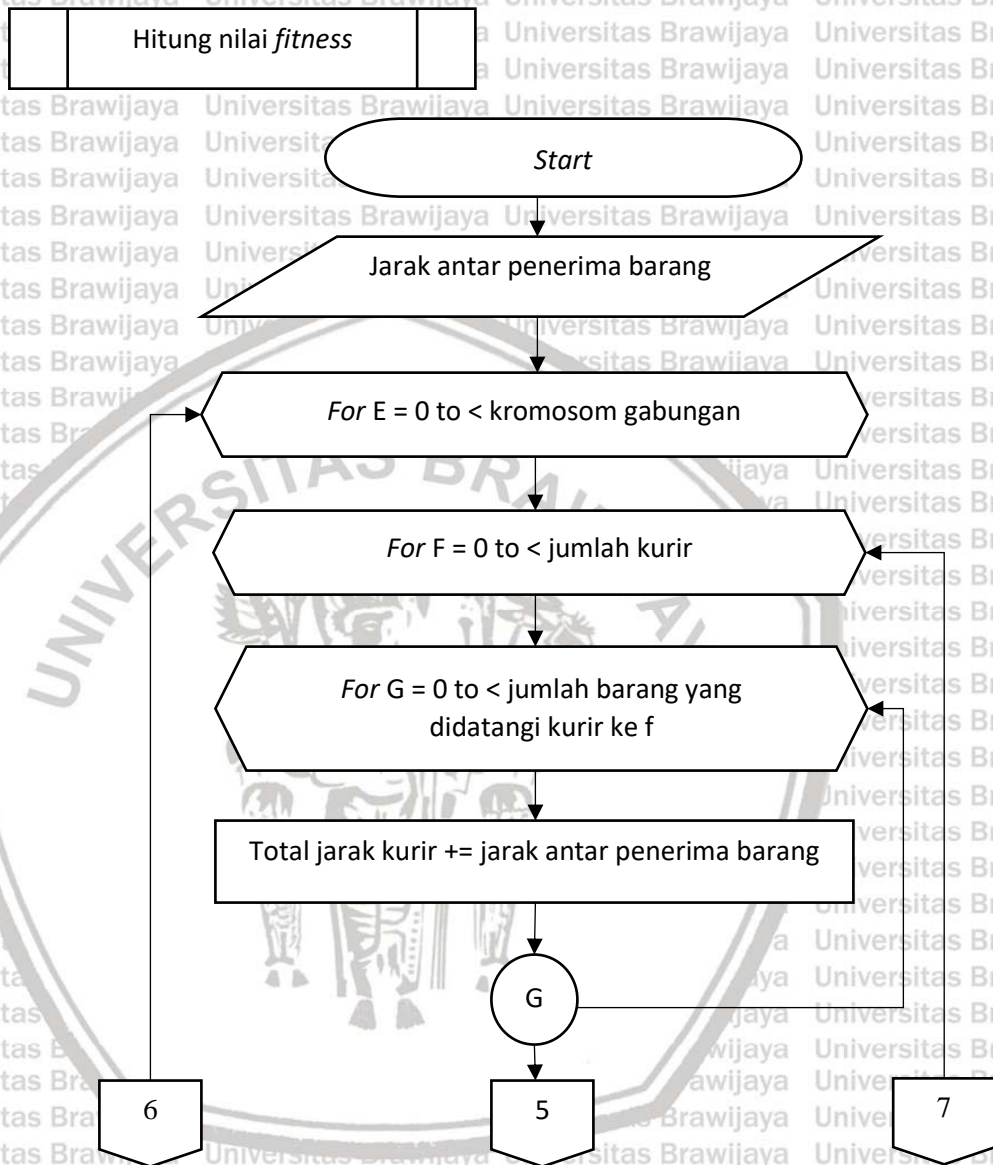
Tabel 4.5 Proses Mutasi

X1														X2			
P3	2	9	12	4	15	7	10	3	14	8	5	11	7	1	13		
C3	2	9	12	11	15	7	10	3	14	8	5	4	7	1	13		

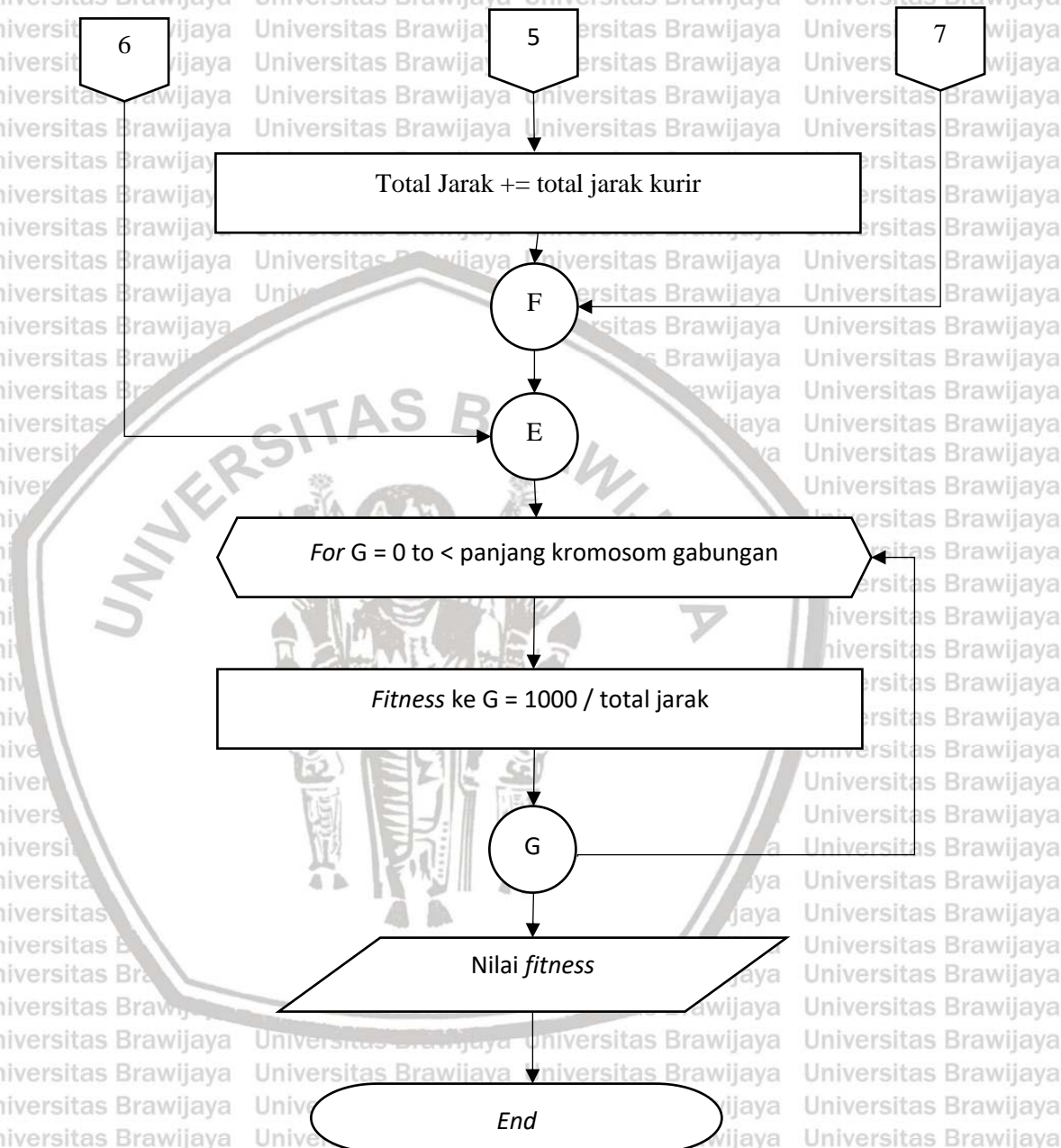
Pada proses mutasi pada Tabel 4.5 ini dimulai dengan menentukan dua posisi gen secara acak. Lalu nilai dari kedua gen dilakukan proses pertukaran lokasi. Setelah nilai gen ditukar maka data tersebut dipindah pada tabel *child*. Tabel *child* ini yang selanjutnya akan digunakan pada proses selanjutnya.

4.3.4 Perhitungan Nilai *Fitness*

Proses perhitung nilai *fitness* merupakan proses untuk mencari nilai *fitness* terbaik dari masing-masing individu. Sebelum melakukan perhitungan nilai *fitness* yaitu menentukan total jarak yang sudah dilewati oleh masing-masing individu. Alur diagram untuk menghitung nilai *fitness* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Hitung Nilai Fitness



Gambar 4.7 Diagram Alir Hitung Nilai *Fitness* (Lanjutan)

Pada diagram di Gambar 4.7 dapat dideskripsikan setiap proses dari nilai *Fitness* sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah data jarak antar penerima barang.

2. Melakukan perulangan variabel $e=0$ hingga kurang dari ((panjang *dataset*)-kromosom gabungan)
3. Melakukan perulangan variabel $f=0$ hingga kurang dari ((panjang *dataset*)-jumlah kurir)
4. Melakukan perulangan variabel $g=0$ hingga kurang dari ((panjang *dataset*)-jumlah data yang didatangi kurir ke)
5. Total jarak kurir += jarak antar penerima barang
6. Loop pada variabel g (langkah 4) hingga mencapai batas jumlah data yang didatangi kurir ke
7. Total jarak += total jarak kurir
8. Loop pada variabel f (langkah 3) hingga data jumlah kurir habis
9. Loop pada variabel e (langkah 2) hingga data kromosom gabungan habis
10. Melakukan perulangan variabel $g=0$ hingga kurang dari ((panjang *dataset*)-panjang kromosom gabungan)
11. $Fitness$ ke $g = 1000 / total\ jarak$
12. Loop pada variabel g (langkah 10) hingga data panjang kromosom gabungan habis
13. Didapatkan *output* nilai *fitness*

Untuk mengetahui *fitness* dari *represent* kromosom pada Tabel 4.2 dan data jarak pada Tabel 2.1 sebagai berikut:

$$Fitness = \frac{1000}{c}$$

Ket:

$c = cost$ (total jarak keseluruhan yang dikunjungi oleh semua kurir)

Total jarak(P1)= 1,1 + 3,2 + 1,1 + 4,0 + 11,9 + 2,1 + 8,2 + 3,1 + 11,6 + 7,8 + 6,9 + 2,9 + 10,1 + 2,2

=76,2

$$Fitness = \frac{1000}{76,2} = 13,12335958$$

Berikut adalah hasil perhitungan manualisasi nilai *fitness* berdasarkan data pada Tabel 4.6 dan hasil secara detail terdapat pada Lampiran C:

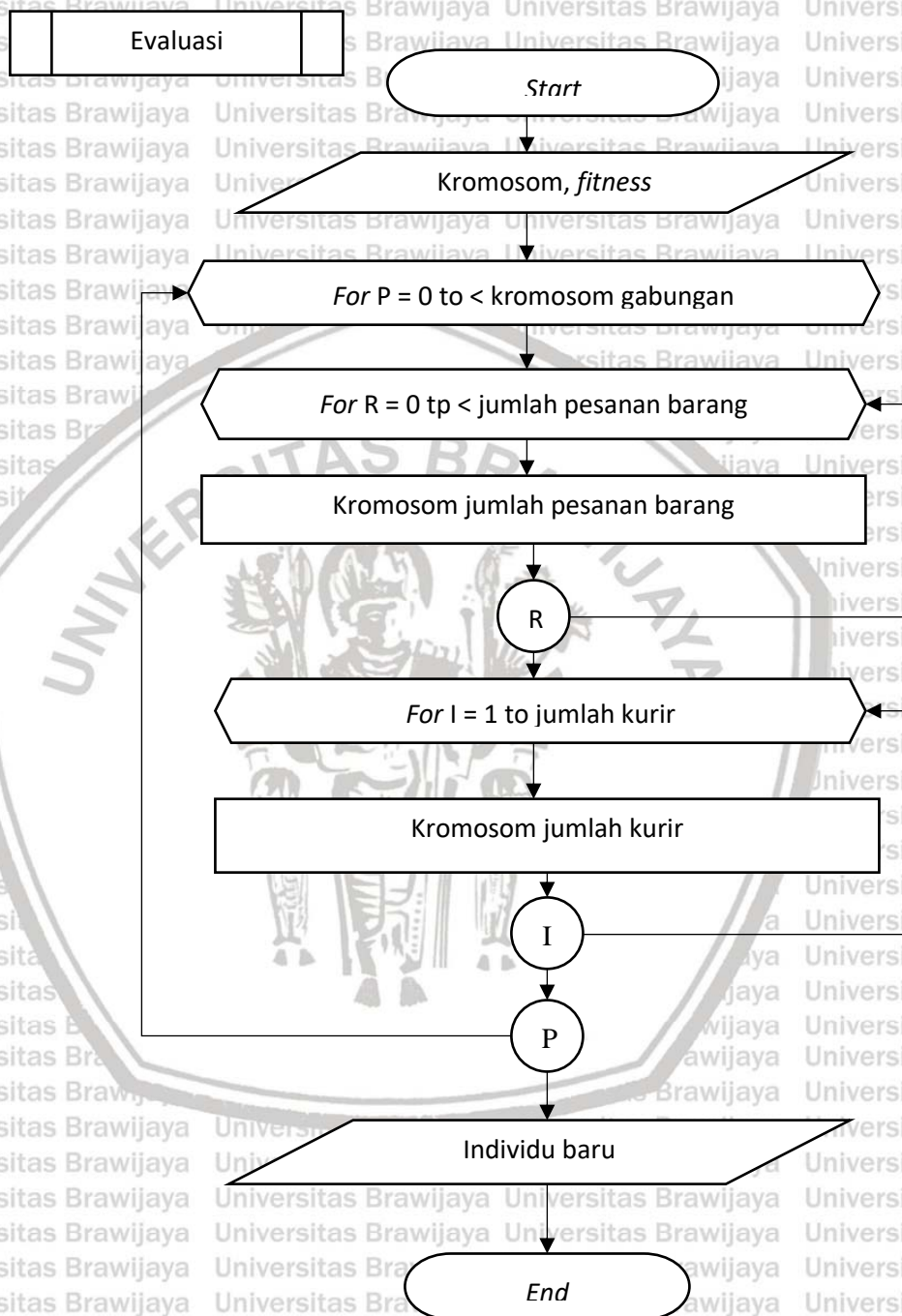
Tabel 4.6 Perhitungan Nilai *Fitness*

P	Kromosom						Total Jarak	Fitness
	Urutan Pengiriman							
	1	2	3	...	14	15		
P1	3	11	7	...	9	6	76.2	13.123
		1.1	3.2	...	10.1	2.2		
P2	5	2	9	...	10	4	83.7	11.947
		3.0	3.9	...	5.2	3.9		
P3	7	4	6	...	9	14	88.5	11.299
		11.0	1.2	...	10.1	13.3		
P4	2	9	12	...	1	13	68.2	14.662
		3.9	5.0	...	2.4	4.0		
P5	12	4	14	...	5	2	78.9	12.674
		7.8	6.7	...	1.8	3.0		
C1	3	11	7	...	8	4	77.8	12.853
		1.1	3.2	...	11.6	2.7		
C2	5	2	9	...	12	4	75.9	13.175
		3.0	3.9	...	11.6	7.8		
C3	2	9	12	...	1	13	96.6	10.351
		3.9	5.0	...	2.4	4.0		

Proses perhitungan nilai *fitness* pada Tabel 4.6 menjelaskan nilai *fitness* didapatkan dari menghitung nilai seluruh nilai jarak berdasarkan urutan *popsi*. Dimana jarak pada setiap *popsi* ini didapatkan dari hasil acak yang dilakukan pada proses sebelumnya. Setelah seluruh jarak dijumlahkan maka akan dihitung nilai *fitness* menggunakan rumus *fitness*. Kemudian didapatkan nilai *fitness* berdasarkan urutan *popsi*. Nilai *fitness* terbaik terdapat pada nilai *fitness* tertinggi.

4.3.5 Evaluasi dan Seleksi

Evaluasi adalah langkah untuk mengumpulkan individu awal dan individu baru. Dimana terdapat campuran populasi yang terbentuk dari individu awal (*parent*) dan individu baru (*child*), kemudian menjadi satu populasi. Diagram alir evaluasi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Diagram Alir Evaluasi

Pada diagram di Gambar 4.8 dapat dideskripsikan setiap proses dari evaluasi sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah kromosom dan nilai *fitness*.
2. Melakukan perulangan variabel $p=0$ hingga kurang dari ((panjang *dataset*)-kromosom gabungan)
3. Melakukan perulangan variabel $r=0$ hingga kurang dari ((panjang *dataset*)-pesanan barang)
4. Kromosom jumlah pesanan barang
5. *Loop* pada variabel r (langkah 3) hingga mencapai batas pesanan barang habis
6. Melakukan perulangan variabel $l=0$ hingga kurang dari ((panjang *dataset*)-jumlah kurir)
7. Kromosom jumlah kurir
8. *Loop* pada variabel l (langkah 6) hingga data jumlah kurir habis
9. *Loop* pada variabel p (langkah 2) hingga data kromosom gabungan habis
10. Didapatkan *output* individu baru

Berikut adalah hasil perhitungan manualisasi evaluasi berdasarkan data pada Tabel

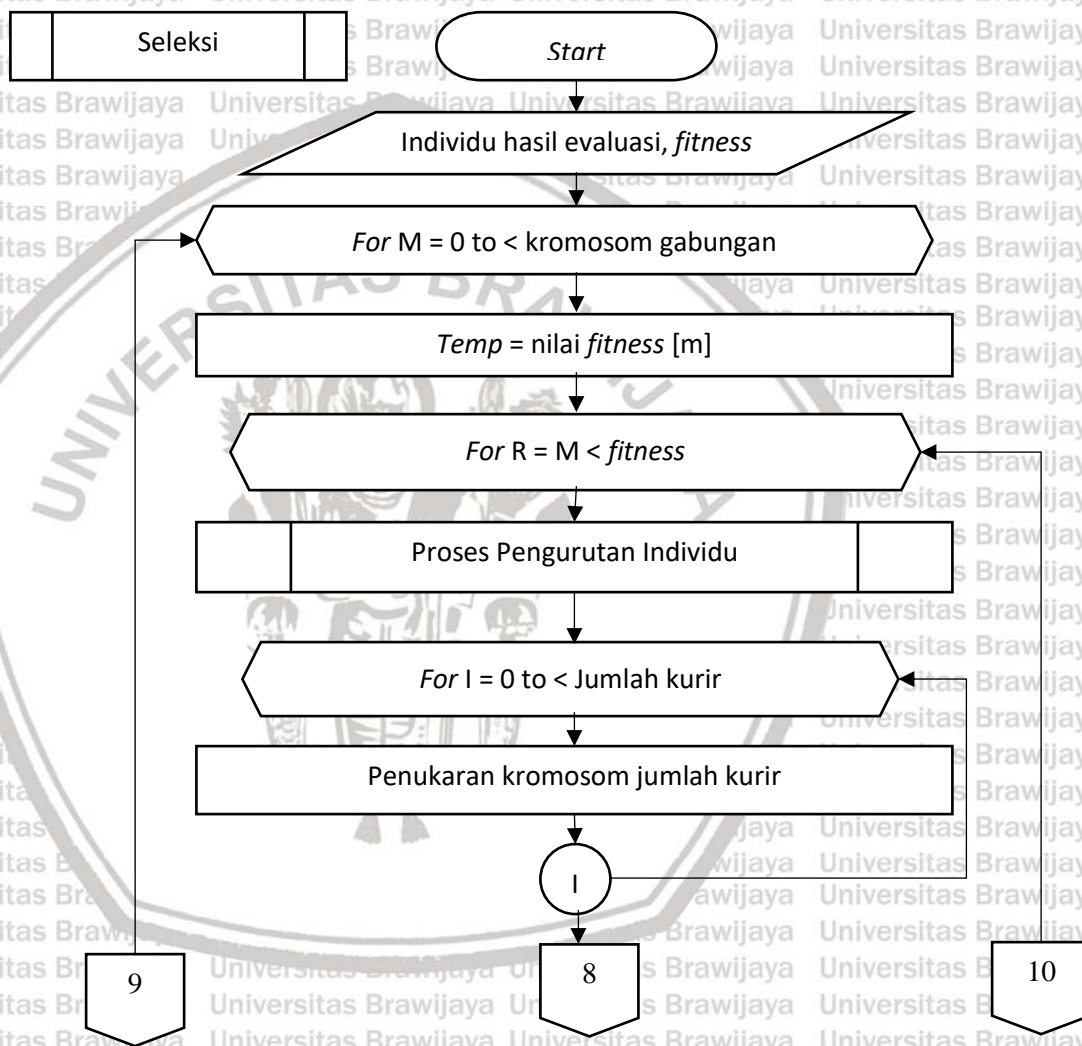
4.1:

Tabel 4.7 Hasil Evaluasi

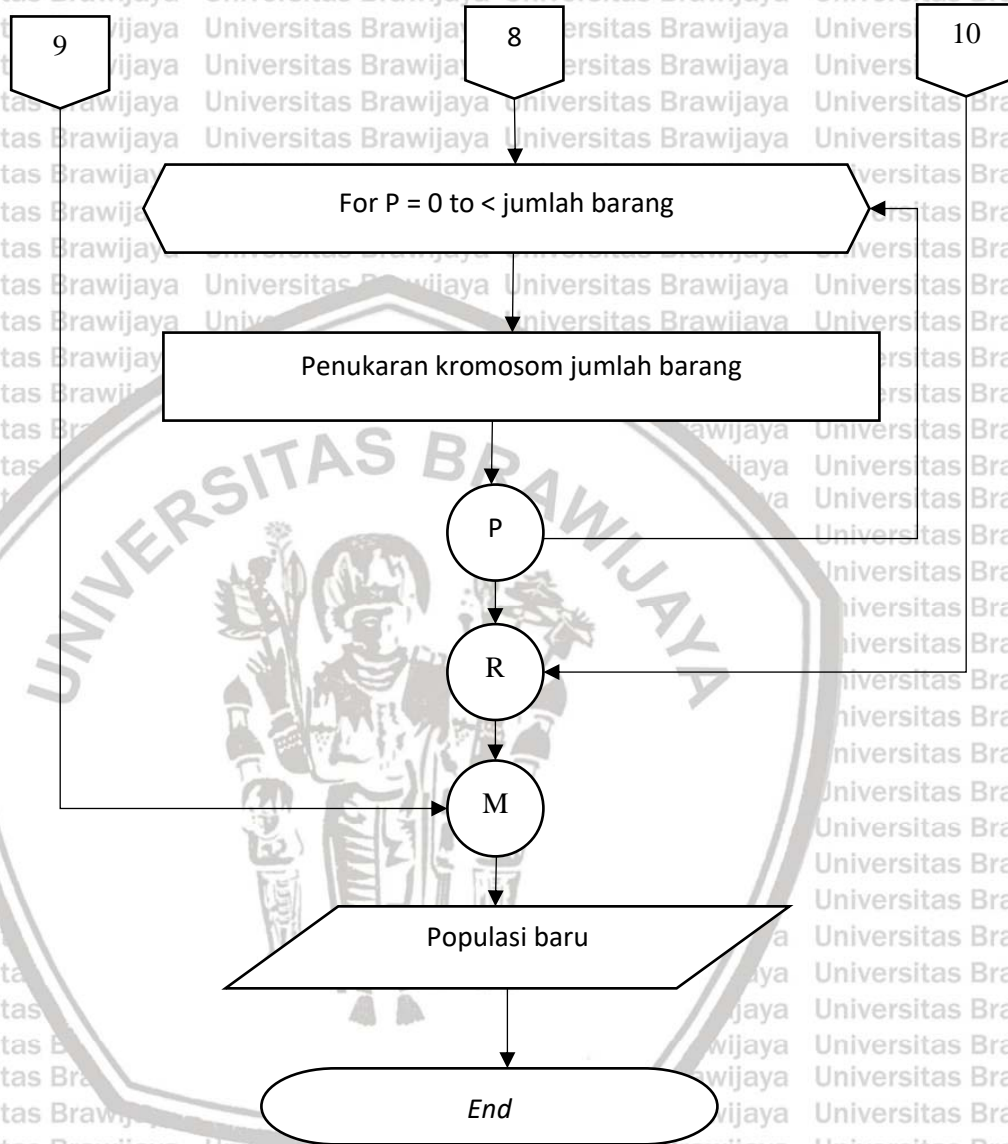
	Kromosom															Fitness
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6	13.12335958
P2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4	11.9474313
P3	7	4	6	11	13	8	2	5	10	12	1	3	15	9	14	11.29943503
P4	2	9	12	4	15	7	10	3	14	8	5	11	7	1	13	14.6627565
P5	12	4	14	11	8	13	7	10	1	15	6	9	3	5	2	12.67427123
C1	3	11	7	1	13	10	5	2	9	15	6	14	12	8	4	12.85347044
C2	5	2	9	15	11	6	13	3	7	1	10	14	8	12	4	13.17523057
C3	2	9	12	11	15	7	10	3	14	8	5	4	7	1	13	10.3519668

Pada hasil evaluasi Tabel 4.7 menjelaskan bahwa proses evaluasi dilakukan dengan mengumpulkan individu awal (*parent*) dan individu baru (*child*) dalam satu tabel. Dimana semua individu tersebut sudah dihitung nilai *fitness* terlebih dahulu. Sehingga akan terdapat nilai *fitness* dari individu lama dan individu baru.

Seleksi merupakan proses terakhir setelah proses evaluasi. Proses seleksi merupakan proses dimana memilih individu untuk proses persilangan sehingga diperoleh calon induk yang baik. Diagram alir seleksi dapat dilihat pada Gambar 4.9.



Gambar 4.9 Diagram Alir Seleksi



Gambar 4.9 Diagram Alir Seleksi (Lanjutan)

Pada diagram di Gambar 4.9 dapat dideskripsikan setiap proses dari seleksi sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah individu hasil evaluasi dan *fitness*.
2. Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-kromosom gabungan)
3. Hitung $Temp = nilai\ fitness[m]$

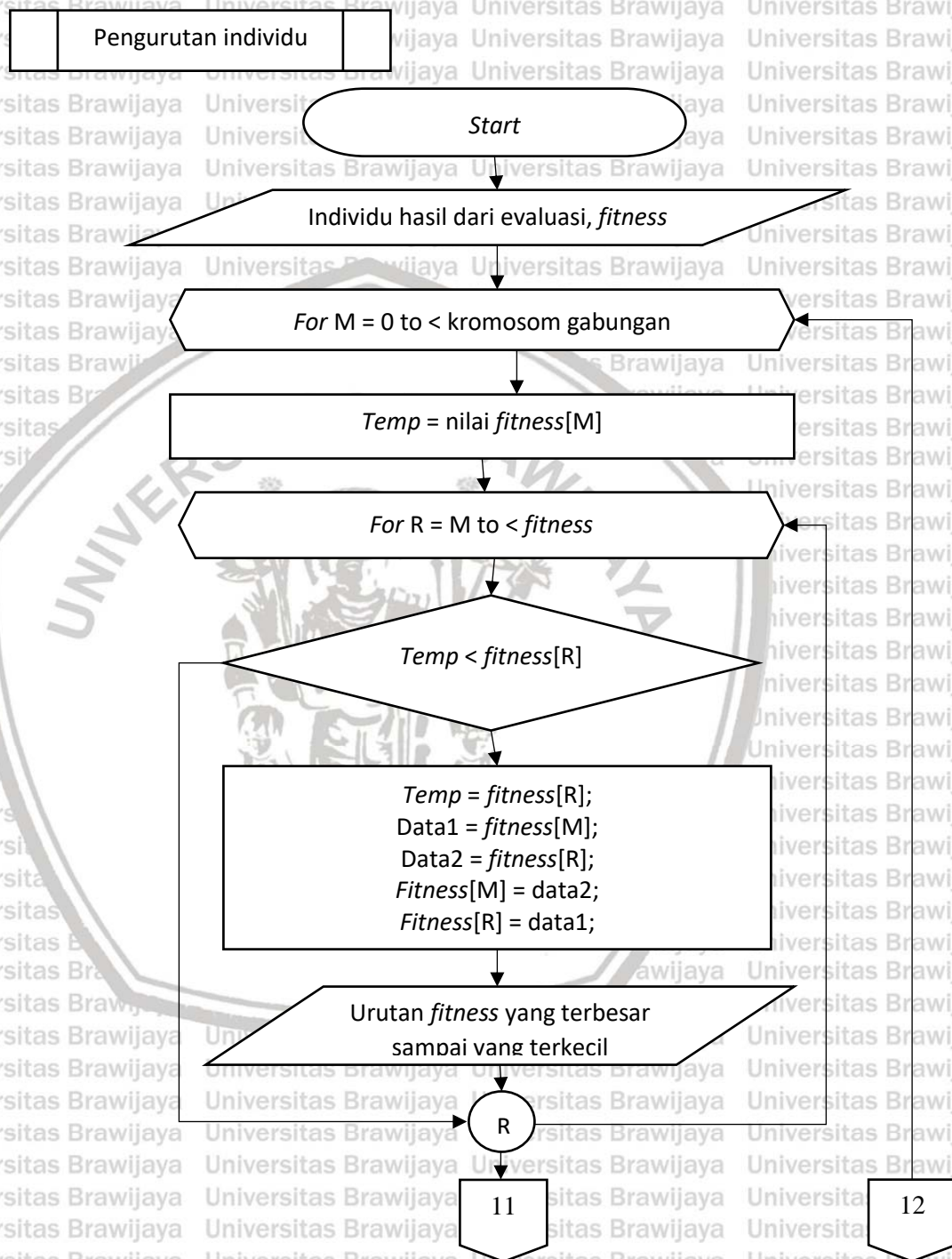
4. Melakukan perulangan variabel $r=m$ hingga kurang dari ((panjang *dataset*)-*fitness*)
5. Memanggil proses pengurutan individu
6. Melakukan perulangan variabel $l=0$ hingga kurang dari ((panjang *dataset*)-jumlah kurir)
7. Pertukaran kromosom jumlah kurir
8. *Loop* pada variabel l (langkah 6) hingga mencapai batas jumlah kurir habis
9. Melakukan perulangan variabel $p=0$ hingga kurang dari ((panjang *dataset*)-jumlah barang)
10. Penukaran kromosom jumlah barang
11. *Loop* pada variabel p (langkah 9) hingga data jumlah barang habis
12. *Loop* pada variabel r (langkah 4) hingga data *fitness* habis
13. *Loop* pada variabel m (langkah 2) hingga data kromosom gabungan habis
14. Didapatkan *output* populasi baru

Berikut adalah hasil perhitungan manualisasi proses seleksi berdasarkan data pada Tabel 4.7:

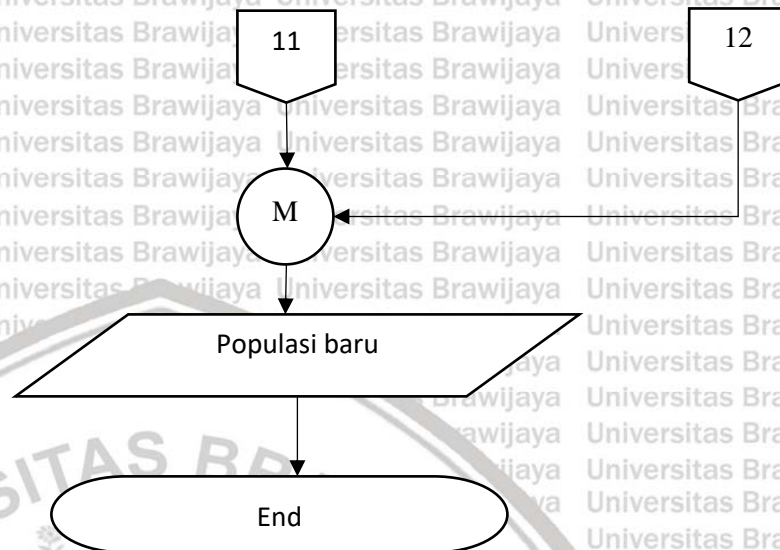
Tabel 4.8 Hasil Seleksi

	Kromosom															Fitness
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6	13.12335958
P2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4	11.9474313
P3	7	4	6	11	13	8	2	5	10	12	1	3	15	9	14	11.29943503
P4	2	9	12	4	15	7	10	3	14	8	5	11	7	1	13	14.6627565
P5	12	4	14	11	8	13	7	10	1	15	6	9	3	5	2	12.67427123
C1	3	11	7	1	13	10	5	2	9	15	6	14	12	8	4	12.85347044
C2	5	2	9	15	11	6	13	3	7	1	10	14	8	12	4	13.17523057
C3	2	9	12	11	15	7	10	3	14	8	5	4	7	1	13	10.3519668

Pada hasil seleksi Tabel 4.8 menjelaskan bahwa proses ini dilakukan dengan melakukan seleksi urutan kromosom berdasarkan besar nilai *fitness* yang didapatkan. Dimana kromosom dengan nilai tertinggi akan terpilih sedangkan kromosom dengan nilai terendah akan tidak terpilih. Sehingga pada Tabel 4.8 untuk kromosom dengan nilai *fitness* terendah diblok dengan warna kuning.



Gambar 4.10 Diagram Alir Pengurutan Individu



Gambar 4.10 Diagram Alir Pengurutan Individu (Lanjutan)

Pada diagram di Gambar 4.10 dapat dideskripsikan setiap proses dari proses pengurutan individu sebagai berikut:

1. Memasukkan *input file dataset* untuk optimasi pengiriman barang, data yang dimasukkan adalah individu hasil evaluasi dan *fitness*.
2. Melakukan perulangan variabel $m=0$ hingga kurang dari ((panjang *dataset*)-kromosom gabungan)
3. Hitung $Temp = \text{nilai } fitness[m]$
4. Melakukan perulangan variabel $r=m$ hingga kurang dari ((panjang *dataset*)-*fitness*)
5. Melakukan pengujian apakah $temp$ lebih kecil dari $fitness[r]$
6. Jika iya maka $temp = fitness[r]$; $data1 = fitness[m]$; $data2 = fitness[r]$; $fitness[m] = data2$; $fitness[r] = data1$.
7. Urutan *fitness* yang terbesar sampai yang terkecil
8. *Loop* pada variabel r (langkah 4) hingga mencapai batas *fitness* habis
9. *Loop* pada variabel m (langkah 2) hingga data kromosom gabungan habis
10. Didapatkan *output* populasi baru

Berikut adalah hasil perhitungan manualisasi proses pengurutan individu berdasarkan data pada Tabel 4.8:

Tabel 4.9 Hasil Proses Pengurutan Individu

	Kromosom															Fitness
P4	2	9	12	4	15	7	10	3	14	8	5	11	7	1	13	14.6627565
C2	5	2	9	15	11	6	13	3	7	1	10	14	8	12	4	13.17523057
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6	13.12335958
C1	3	11	7	1	13	10	5	2	9	15	6	14	12	8	4	12.85347044
P5	12	4	14	11	8	13	7	10	1	15	6	9	3	5	2	12.67427123
P2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4	11.9474313
P3	7	4	6	11	13	8	2	5	10	12	1	3	15	9	14	11.29943503
C3	2	9	12	11	15	7	10	3	14	8	5	4	7	1	13	10.3519668

Pada hasil proses pengurutan individu Tabel 4.9 menjelaskan proses dilakukan setelah proses seleksi berhasil dilakukan. Setelah berhasil menyeleksi data kromosom dengan nilai *fitness* tertinggi dan terendah maka kemudian dilakukan pengurutan individu. Pengurutan ini dilakukan berdasarkan nilai *fitness* dari nilai tertinggi hingga terendah. Setelah dilakukan pengurutan individu maka pengguna akan dengan mudah memilih rute pengiriman barang yang paling optimal berdasarkan nilai *fitness* yang didapatkan.

4.4 Perancangan Antarmuka

Pada bagian untuk perancangan antarmuka sistem bertujuan untuk menjelaskan rancangan dari tampilan implementasi untuk penelitian tentang optimasi pengiriman barang menggunakan Algoritme Genetika. Perancangan antarmuka penelitian ini terdapat 3 halaman untuk menampilkan Halaman Data, Halaman Parameter, Halaman Algoritme Genetika. Memiliki 2 *dataset* yaitu, data barang dan data kurir.

4.4.1 Perancangan Halaman Data

Pada halaman rancangan antarmuka ini sebagai halaman yang akan menampilkan saat pertama kali dijalankan. Halaman data ini berfungsi untuk memasukkan data yang dibutuhkan dalam penelitian ini. Berikut adalah rancangan antarmuka halaman data:

**1 OPTIMASI PENGIRIMAN BARANG MENGGUNAKAN
ALGORITMA GENETIKA**

Data **2** Parameter **3** Algoritma Genetika **4**

Data Pengirim **5**

No	Nama	Alamat

Data Kurir **6**

ID	Nama

Gambar 4.11 Perancangan Antarmuka Halaman Data

Keterangan:

1. Nama judul dari penelitian.
2. *Button tab* data berfungsi sebagai membuka tampilan halaman data.
3. *Button tab* parameter berfungsi sebagai membuka tampilan halaman parameter output.
4. *Button tab* algoritme genetika berfungsi sebagai menampilkan rute hasil perhitungan algoritme genetika.
5. Kolom yang berfungsi untuk menampilkan hasil proses algoritma (Data Pengirim).
6. Kolom yang berfungsi untuk menyajikan hasil dari proses algoritme (Data Kurir).

4.4.2 Perancangan Halaman Parameter

Berikut rancangan antarmuka halaman parameter terdapat pada Gambar 4.12.

1 OPTIMASI PENGIRIMAN BARANG MENGGUNAKAN

ALGORITMA GENETIKA

Data **2**

Parameter **3**

Algoritma Genetika **4**

Input Pengirim

No. Pengirim:

Nama Pengirim:

Add Pengirim **7**

Input Kurir

No. ID:

Nama Kurir:

Add Kurir **10**

Proses **11**

Populasi Baru:

Kurir A (Nama Kurir) :

Nama Pengirim1 – Nama Pengirim2 – Nama Pengirim3 –

Jarak Tempuh :

Kurir B (Nama Kurir) :

Nama Pengirim1 – Nama Pengirim2 – Nama Pengirim3 –

Jarak Tempuh :

Total Jarak Tempuh :

12

Gambar 4.12 Perancangan Antarmuka Halaman Parameter

Keterangan:

1. Nama judul dari penelitian.
2. *Button tab* data berfungsi sebagai membuka tampilan halaman data.
3. *Button tab* parameter berfungsi sebagai membuka tampilan halaman parameter *output*.
4. *Button tab* algoritme genetika berfungsi sebagai menampilkan rute hasil perhitungan algoritme genetika.
5. Kolom yang berfungsi untuk mengisi data yang akan diinputkan (No. Pengirim).
6. Kolom yang berfungsi untuk mengisi data yang akan diinputkan (Nama Pengirim).
7. *Button Add Pengirim* berfungsi untuk menambahkan data pengirim
8. Kolom yang berfungsi untuk mengisi data yang akan diinputkan (No. ID).
9. Kolom yang berfungsi untuk mengisi data yang akan diinputkan (data barang).

10. *Button Add Kurir* berfungsi untuk Menambahkan data kurir.
11. *Button* proses berfungsi untuk menjalankan proses algoritme genetika pada data yang telah dimasukkan.
12. Kolom yang berfungsi untuk menampilkan populasi baru berdasarkan data yang diinputkan.

4.4.3 Perancangan Halaman Algoritme Genetika

Berikut rancangan antarmuka halaman algoritme genetika terdapat pada Gambar 4.13.

1

OPTIMASI PENGIRIMAN BARANG MENGGUNAKAN

ALGORITMA GENETIKA

Data 2

Parameter 3

Algoritma genetika 4

OUTPUT

Cr:

5

Mr:

6

Popsizen:

7

Gen:

8

Set 9

Urutan Pengiriman	Kurir	Total Jarak	Nilai Fitness
10			

Gambar 4.13 Perancangan Antarmuka Halaman Algoritme Genetika

Keterangan:

1. Nama judul dari penelitian.
2. *Button tab* data berfungsi sebagai membuka tampilan halaman data.

3. *Button tab* parameter berfungsi sebagai membuka tampilan halaman parameter *output*.
4. *Button tab* algoritme genetika berfungsi sebagai menampilkan rute hasil perhitungan algoritme genetika.
5. Kolom yang berfungsi untuk menampilkan hasil proses algoritme (nilai Cr).
6. Kolom yang berfungsi untuk menampilkan hasil proses algoritme (nilai Mr).
7. Kolom yang berfungsi untuk menampilkan hasil proses algoritme (*Popsiz*e).
8. Kolom yang berfungsi untuk menampilkan hasil proses algoritme (Gen).
9. *Button set* berfungsi sebagai menampilkan hasil *output* pada kolom outputnya masing – masing.
10. Kolom yang berfungsi untuk menampilkan hasil proses algoritme (*Output Hasil Iterasi*).

4.5 Perancangan Pengujian

Pada bagian untuk perancangan pengujian yang akan dilakukan untuk Algoritme Genetika bertujuan agar dapat mengetahui nilai parameter yang paling optimal pada optimasi pengiriman barang menggunakan Algoritme Genetika. Perancangan pengujian yang digunakan pada proses pengujian nilai parameter pada Algoritme Genetika adalah sebagai berikut:

1. Uji coba ukuran populasi
2. Uji coba banyak generasi
3. Uji coba kombinasi Cr & Mr

Dari skenario pengujian diatas, penulis melakukan proses pengujian yang berulang berdasar dari nilai parameter yang terbaik yang didapatkan sebelumnya. Pada setiap nilai parameter pengujian akan dibuat percobaan sebanyak 10 percobaan. Percobaan dilakukan sebanyak 10 kali dikarenakan angka tersebut sudah dapat mewakili jumlah percobaan yang dapat dilakukan untuk mendapatkan hasil yang optimal.

4.5.1 Uji Coba Ukuran Populasi

Pengujian ukuran populasi dilakukan agar dapat memperoleh ukuran populasi pada setiap percobaan yang telah dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan.

Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap – tiap percobaan yang didapatkan per ukuran populasi. Dan juga didapatkan nilai rata – rata *fitness* dari semua nilai percobaan yang didapatkan pada per ukuran populasi. Perancangan uji coba ukuran populasi dapat dilihat pada Tabel 4.10.

Tabel 4.10 Perancangan Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai Fitness Percobaan ke - n										Fitness tertinggi	Rata – rata fitness
	1	2	3	4	5	6	7	8	9	10		
10												
20												
30												
40												
50												
60												
70												
80												
90												
100												

4.5.2 Uji Coba Banyak Generasi

Pada bagian pengujian banyak generasi ini dilakukan untuk mengetahui banyak generasi pada setiap percobaan yang telah dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan. Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap – tiap percobaan yang didapatkan per banyak generasi. Dan juga didapatkan nilai rata – rata *fitness* dari semua nilai percobaan yang didapatkan pada per banyak generasi. Perancangan uji coba banyak generasi dapat dilihat pada Tabel 4.11.

Tabel 4.11 Perancangan Uji Coba Banyak Generasi

Banyak Generasi	Nilai Fitness Percobaan ke - n										Fitness tertinggi	Rata – rata fitness
	1	2	3	4	5	6	7	8	9	10		
50												
100												

4.5.3 Uji Coba Kombinasi Cr & Mr

Tabel 4.12 Perancangan Uji Coba Kombinasi Cr & Mr

59

4.6 Uji Coba Analisis Global

Pada bagian pengujian sistem ini dijalankan dengan melakukan perbandingan hasil perhitungan manualisasi dengan hasil perhitungan dari sistem. Dimana hasil perhitungan manualisasi merupakan hasil pengiriman yang dilakukan oleh JNE tanpa bantuan sistem. Perancangan uji coba perhitungan pengiriman manualisasi dapat dilihat pada Tabel 4.13.

Tabel 4.13 Perancangan Uji Coba Perhitungan Pengiriman Manualisasi

Nama kurir	Kunjungan	Total jarak
Nama kurir 1	Pelanggan 1 – pelanggan 2 – pelanggan 3....	
Nama kurir 2		
Nama kurir 3		

Selanjutnya uji coba yang dilakukan adalah untuk mengetahui Sistem berjalan pada setiap percobaan dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan. Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap-tiap percobaan yang didapatkan per uji coba Sistem. Dan juga didapatkan nilai rata-rata *fitness* dari semua nilai percobaan yang didapatkan pada per uji coba Sistem. Perancangan uji coba Sistem ini dapat dilihat pada Tabel 4.14.

Tabel 4.14 Perancangan Uji Coba Sistem

Nama kurir	Kunjungan	Total jarak
Nama kurir 1	Pelanggan 1 – pelanggan 2 – pelanggan 3....	
Nama kurir 2		
Nama kurir 3		

Terakhir, uji coba yang dilakukan adalah untuk mengetahui hasil perbandingan rekomendasi sistem dan manualisasi. Perancangan uji coba hasil perbandingan rekomendasi sistem dan manualisasi ini dapat dilihat pada Tabel 4.15.

Tabel 4.15 Perancangan Uji Coba Perbandingan Rekomendasi Sistem Dan Manualisasi

Kurir	Kurir 1 (KM)	Kurir 2 (KM)	Kurir 3 (KM)	Kurir 4 (KM)	Total jarak	Fitness
-------	-----------------	-----------------	-----------------	-----------------	----------------	---------

Hasil pengiriman JNE						
Hasil Rekomendasi sistem						
Selisih						



BAB 5 IMPLEMENTASI

5.1 Implementasi Program

Pada bagian bab ini akan dilakukan penerapan dari perancangan sesuai dengan yang telah dibuat sebelumnya di rancangan algoritme genetika. Dalam implementasi ini akan dibagi menjadi beberapa proses implementasi menjadi kode program algoritma genetika yaitu kode program pengambilan data, kode program *input* parameter, kode program inialisasi populasi awal, kode program reproduksi, kode program total jarak dan *fitness*, kode program evaluasi dan seleksi. Pada setiap proses akan memiliki sub proses kode program sesuai dengan perancangan Algoritme Genetika.

5.2 Proses Pengambilan Data

Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan algoritma genetika secara umum dari pengambilan data yang dibutuhkan seperti data barang, data kurir, dan data jarak pengiriman barang. Kode program untuk implementasi fungsi input data terdapat pada Kode Program 5.1.

Algoritma 1: Fungsi <i>Input</i> Data	
1	<code>void showData()</code>
2	<code>{</code>
3	<code> DataPanel pn=new DataPanel();</code>
4	<code> pn.setLocation(0,0);</code>
5	<code> pnlTengah.removeAll();</code>
6	<code> //JLabel lb=new JLabel();</code>
7	<code> //lb.setText("bb");</code>
8	<code> //pnlTengah.add(lb);</code>
9	<code> //pnlTengah.invalidate();</code>
10	<code> javax.swing.GroupLayout pnlTengahLayout = new</code>
11	<code> javax.swing.GroupLayout(pnlTengah);</code>
12	<code> pnlTengah.setLayout(pnlTengahLayout);</code>
13	<code></code>
14	<code> pnlTengah.setLayout(pnlTengahLayout);</code>
15	<code> pnlTengahLayout.setHorizontalGroup(</code>
16	<code> pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)</code>
17	<code> .addGroup(pnlTengahLayout.createSequentialGroup())</code>
18	<code> .addGap(19, 19, 19)</code>
19	<code> .addComponent(pn)</code>
20	<code> .addContainerGap(452, Short.MAX_VALUE))</code>
21	<code>}</code>

```

22 );
23 pnlTengahLayout.setVerticalGroup(
24
25 pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
26     .addGroup(pnlTengahLayout.createSequentialGroup())
27     .addContainerGap()
28     .addComponent(pn)
29     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
30 );
31 }
32 void showInput()
33 {
34     InputPanel pn=new InputPanel();
35     pn.setLocation(0,0);
36     pnlTengah.removeAll();
37     //JLabel lb=new JLabel();
38     //lb.setText("bb");
39     //pnlTengah.add(lb);
40     //pnlTengah.invalidate();
41     javax.swing.GroupLayout pnlTengahLayout = new
42     javax.swing.GroupLayout(pnlTengah);
43     pnlTengah.setLayout(pnlTengahLayout);
44
45     pnlTengah.setLayout(pnlTengahLayout);
46     pnlTengahLayout.setHorizontalGroup(
47
48     pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
49     .addGroup(pnlTengahLayout.createSequentialGroup())
50     .addGap(19, 19, 19)
51     .addComponent(pn)
52     .addContainerGap(452, Short.MAX_VALUE))
53 );
54     pnlTengahLayout.setVerticalGroup(
55
56     pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
57     .addGroup(pnlTengahLayout.createSequentialGroup())
58     .addContainerGap()
59     .addComponent(pn)
60     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
61 );
62 }
63 void showOutput()
64 {
65     OutputPanel pn=new OutputPanel();
66     pn.setLocation(0,0);
67     pnlTengah.removeAll();
68     //JLabel lb=new JLabel();
69     //lb.setText("bb");
70     //pnlTengah.add(lb);

```



```

71 //pnlTengah.invalidate();
72 javax.swing.GroupLayout pnlTengahLayout = new
73 javax.swing.GroupLayout(pnlTengah);
74 pnlTengah.setLayout(pnlTengahLayout);
75
76 pnlTengah.setLayout(pnlTengahLayout);
77 pnlTengahLayout.setHorizontalGroup(
78
79 pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
80 .addGroup(pnlTengahLayout.createSequentialGroup()
81 .addGap(19, 19, 19)
82 .addComponent(pn)
83 .addGap(452, 452, Short.MAX_VALUE))
84 );
85 pnlTengahLayout.setVerticalGroup(
86
87 pnlTengahLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
88 .addGroup(pnlTengahLayout.createSequentialGroup()
89 .addGap(19, 19, 19)
90 .addComponent(pn)
91 .addGap(452, 452, Short.MAX_VALUE))
92 );
93 }

```

Kode Program 5.1 Pengambilan Data Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.1 adalah sebagai berikut:

1. Baris 1 menjelaskan kelas untuk *show* data
2. Baris 2-28 menjelaskan pengaturan *layout* untuk menampilkan kolom *input* data yang telah dimasukkan
3. Baris 29-56 menjelaskan pengaturan *layout* untuk menampilkan kolom pengisian data yang akan diisi
4. Baris 57-84 menjelaskan pengaturan *layout* untuk menampilkan kolom *output* data dari data *input* yang telah dimasukkan

5.3 Proses *Input* Parameter

Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan Algoritme Genetika secara umum dari input parameter. Kode program untuk implementasi fungsi input parameter terdapat pada Kode Program 5.2.

Algoritma 2: Fungsi *Input* parameter

```

1 public void showDetail(Calon cl)
2 {
3     String hasil="";
4     int start=0;
5     int totalJarakTempuh=0;
6     for (int i=0;i<FormProses.daftarKurir.size();i++)
7     {
8         hasil=hasil+FormProses.daftarKurir.get(i)+":\n";
9         int jumlah=cl.jumlah.get(i);
10        String kirim="";
11        int i1=0;
12        int totalJarak=0;
13        for (int x=start;x<(start+jumlah);x++)
14        {
15            if (!kirim.equals(""))
16            {
17                kirim=kirim+"-";
18            }
19            int idx=Integer.parseInt(cl.calon.get(x));
20            int jarak=Integer.parseInt(FormProses.model.getValueAt(i1,idx+1)+"");
21            totalJarak+=jarak;
22            i1=idx;
23            kirim=kirim+FormProses.daftarPengirim.get(idx);
24        }
25        hasil=hasil+kirim+"\n";
26        hasil=hasil+"Total Jarak "+totalJarak+"\n";
27        hasil=hasil+"\n";
28        start=start+jumlah;
29
30        totalJarakTempuh+=totalJarak;
31    }
32
33    hasil=hasil+"Jarak Total "+totalJarakTempuh;
34    tbResult.setText(hasil);
35    }
36
37    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
38        // TODO add your handling code here:
39        try
40        {
41            //String q="INSERT INTO anggota
42            (Username,Password>Nama,NoKTP,NPWP,TanggalMulaiKerja) VALUES
43            (?,?,'test','ktp','npwp',SYSDATE())";
44            String q="INSERT INTO pengiriman (no_pengirim,nama_pengirim,alamat_pengirim)
45            VALUES (?,?,:)";
46            PreparedStatement stmt =Gbaru.con.prepareStatement(q);
47            stmt.setString(1, etNoPengirim.getText());
48            stmt.setString(2, etNamaPengirim.getText());
49            stmt.setString(3, etAlamat.getText());

```



```

45      stmt.execute();
46
47      etNoPengirim.setText("");
48      etNamaPengirim.setText("");
49      etAlamat.setText("");
50      JOptionPane.showMessageDialog(null, "Pengirim berhasil diinputkan");
51  }
52  catch (Exception ex)
53  {
54      ex.printStackTrace();
55  }
56
57  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
58      // TODO add your handling code here:
59      try
60      {
61          //String q="INSERT INTO anggota
62          (Username,Password>Nama,NoKTP,NPWP,TanggalMulaiKerja) VALUES
63          (?,?,'test','ktp','npwp',SYSDATE())";
64          String q="INSERT INTO kurir (id_k,nama) VALUES (?,?)";
65          PreparedStatement stmt=Gbaru.con.prepareStatement(q);
66          stmt.setString(1, etIDKurir.getText());
67          stmt.setString(2, etNamaKurir.getText());
68          stmt.execute();
69          JOptionPane.showMessageDialog(null, "Kurir berhasil diinputkan");
70          etIDKurir.setText("");
71          etNamaKurir.setText("");
72      }
73      catch (Exception ex)
74      {
75          ex.printStackTrace();
76      }
77  }

```

Kode Program 5.2 Input Parameter Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.2 adalah sebagai berikut:

1. Baris 1 menjelaskan kelas untuk show detail
2. Baris 2-33 menjelaskan untuk perulangan untuk menghitung total jarak dan fitness dari daftar pengirim yang dilakukan oleh kurir

3. Baris 34-54 menjelaskan perulangan untuk menerima data *input* (data pengirim)
4. Baris 55-73 menjelaskan perulangan untuk menerima data *input* (data kurir)

5.3.1 Proses memasukkan Data Pengirim

Pada bagian proses memasukkan data pengirim ini merupakan penjelasan kode program yang digunakan untuk memasukkan data pengirim pada program Algoritme Genetika ini. Kode program untuk proses memasukkan data pengirim terdapat pada Kode Program 5.3.

Algoritma 3: Fungsi Memasukkan data pengirim	
1	<code>private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {</code>
2	<code>// TODO add your handling code here:</code>
3	<code>try</code>
4	<code>{</code>
5	<code>//String q="INSERT INTO anggota</code>
6	<code>(Username>Password>Nama>NoKTP>NPWP>TanggalMulaiKerja) VALUES</code>
7	<code>(?, ?, 'test', 'ktp', 'npwp', SYSDATE());</code>
8	<code>String q="INSERT INTO pengirim (no_pengirim,nama_pengirim,alamat_pengirim)</code>
9	<code>VALUES (?, ?, ?);</code>
10	<code>PreparedStatement stmt =Gbaru.con.prepareStatement(q);</code>
11	<code>stmt.setString(1, etNoPengirim.getText());</code>
12	<code>stmt.setString(2, etNamaPengirim.getText());</code>
13	<code>stmt.setString(3, etAlamat.getText());</code>
14	<code>stmt.execute();</code>
15	<code>etNoPengirim.setText("");</code>
16	<code>etNamaPengirim.setText("");</code>
17	<code>etAlamat.setText("");</code>
18	<code>JOptionPane.showMessageDialog(null, "Pengirim berhasil diinputkan");</code>
19	<code>}</code>
20	<code>catch (Exception ex)</code>
21	<code>{</code>
22	<code>ex.printStackTrace();</code>
23	<code>}</code>

Kode Program 5.3 Memasukkan Data Pengirim

Untuk penjelasan pada implementasi kode program 5.3 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk *input* data pengirim
2. Baris 2-7 menjelaskan untuk memanggil data yang telah dimasukkan
3. Baris 8 menjelaskan untuk mendapatkan data *input* (No Pengirim)
4. Baris 9 menjelaskan untuk mendapatkan data *input* (Nama Pengirim)
5. Baris 10 menjelaskan untuk mendapatkan data *input* (Alamat)
6. Baris 12 menjelaskan untuk mengatur data *input* pada kolom *output* (No Pengirim)
7. Baris 13 menjelaskan untuk mengatur data *input* pada kolom *output* (Nama Pengirim)
8. Baris 14 menjelaskan untuk mengatur data *input* pada kolom *output* (Alamat)
9. Baris 15-21 menjelaskan untuk menampilkan kotak dialog (Pengirim Berhasil Diinputkan)

5.3.2 Proses Memasukkan Data Kurir

Pada bagian proses memasukkan data pengirim ini merupakan penjelasan kode program yang digunakan untuk memasukkan data kurir pada program Algoritme Genetika ini. Kode program untuk proses memasukkan data kurir terdapat pada Kode Program 5.4.

Algoritma 4: Fungsi Memasukkan data kurir	
1	<code>private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {</code>
2	<code>// TODO add your handling code here:</code>
3	<code>try</code>
4	<code>{</code>
5	<code>//String q="INSERT INTO anggota</code>
6	<code>(Username,Password>Nama,NoKTP,NPWP,TanggalMulaiKerja) VALUES</code>
7	<code>(?,?, 'test','ktp','npwp',SYSDATE());</code>
8	<code>String q="INSERT INTO kurir (id_k,nama) VALUES (?,?)";</code>
9	<code>PreparedStatement stmt =Gbaru.con.prepareStatement(q);</code>
10	<code>stmt.setString(1, etIDKurir.getText());</code>
11	<code>stmt.setString(2, etNamaKurir.getText());</code>
12	<code>stmt.execute();</code>
13	<code>JOptionPane.showMessageDialog(null, "Kurir berhasil diinputkan");</code>
14	<code>etIDKurir.setText("");</code>
15	<code>etNamaKurir.setText("");</code>
16	<code>catch (Exception ex)</code>
17	<code>{</code>

```

18      ex.printStackTrace();
19    }
20  }

```

Kode Program 5.4 Memasukkan Data Kurir

Untuk penjelasan pada implementasi kode program 5.4 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk *input* data kurir
2. Baris 2-7 menjelaskan untuk memanggil data yang telah dimasukkan
3. Baris 8 menjelaskan untuk mendapatkan data *input* (ID Kurir)
4. Baris 9 menjelaskan untuk mendapatkan data *input* (Nama Kurir)
5. Baris 11 menjelaskan untuk menampilkan kotak dialog (Kurir Berhasil Diinputkan)
6. Baris 12 menjelaskan untuk mengatur data *input* pada kolom *output* (ID Kurir)
7. Baris 13-19 menjelaskan untuk mengatur data *input* pada kolom *output* (Nama Kurir)

5.3.3 Proses Memasukkan Data Parameter Algoritme Genetika

Proses input data pengirim ini adalah penjelasan kode program yang digunakan untuk memasukkan data parameter pada program Algoritme Genetika ini. Kode program untuk proses memasukkan data parameter Algoritme Genetika terdapat pada Kode Program 5.5.

Algoritma 5:	Fungsi Memasukkan data parameter algoritma genetika
1	<code>private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {</code>
2	<code>// TODO add your handling code here:</code>
3	<code>try</code>
4	<code>{</code>
5	<code>//String q="INSERT INTO anggota</code>
6	<code>(Username,Password>Nama,NoKTP,NPWP,TanggalMulaiKerja) VALUES</code>
7	<code>(?,?, 'test', 'ktp', 'npwp', SYSDATE());</code>
8	<code>String q="INSERT INTO pengirim (no_pengirim,nama_pengirim,alamat_pengirim)</code>
9	<code>VALUES (?,?,?);</code>
10	<code>PreparedStatement stmt =Gbaru.con.prepareStatement(q);</code>
11	<code>stmt.setString(1, etNoPengirim.getText());</code>
12	<code>stmt.setString(2, etNamaPengirim.getText());</code>
13	<code>stmt.setString(3, etAlamat.getText());</code>
14	<code>stmt.execute();</code>
15	<code>etNoPengirim.setText("");</code>


```

15      etNamaPengirim.setText("");
      etAlamat.setText("");
      JOptionPane.showMessageDialog(null, "Pengirim berhasil diinputkan");
16
17
18  }
19  catch (Exception ex)
20  {
21      ex.printStackTrace();
22  }
23
24  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
25      // TODO add your handling code here:
26      try
27      {
28          //String q="INSERT INTO anggota
29          (Username,Password>Nama,NoKTP,NPWP,TanggalMulaiKerja) VALUES
30          (?,?,'test','ktp','npwp',SYSDATE())";
31          String q="INSERT INTO kurir (id_k,nama) VALUES (?,?)";
32          PreparedStatement stmt =Gbaru.con.prepareStatement(q);
33          stmt.setString(1, etIDKurir.getText());
34          stmt.setString(2, etNamaKurir.getText());
35          stmt.executeUpdate();
36
37          JOptionPane.showMessageDialog(null, "Kurir berhasil diinputkan");
38
39          etIDKurir.setText("");
40          etNamaKurir.setText("");
41      }
42      catch (Exception ex)
43      {
44          ex.printStackTrace();
45      }
46  }

```

Kode Program 5.5 Memasukkan Data Parameter Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.5 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk input data pengirim
2. Baris 2-7 menjelaskan untuk memanggil data yang telah dimasukkan
3. Baris 8 menjelaskan untuk mendapatkan data input (No Pengirim)
4. Baris 9 menjelaskan untuk mendapatkan data input (Nama Pengirim)
5. Baris 10 menjelaskan untuk mendapatkan data input (Alamat)

6. Baris 12 menjelaskan untuk mengatur data input pada kolom output (No Pengirim)
7. Baris 13 menjelaskan untuk mengatur data input pada kolom output (Nama Pengirim)
8. Baris 14 menjelaskan untuk mengatur data input pada kolom output (Alamat)
9. Baris 15-21 menjelaskan untuk menampilkan kotak dialog (Pengirim Berhasil Diinputkan)
10. Baris 22 menjelaskan nama kelas untuk input data kurir
11. Baris 23-28 menjelaskan untuk memanggil data yang telah dimasukkan
12. Baris 29 menjelaskan untuk mendapatkan data input (ID Kurir)
13. Baris 30 menjelaskan untuk mendapatkan data input (Nama Kurir)
14. Baris 32 menjelaskan untuk menampilkan kotak dialog (Kurir Berhasil Diinputkan)
15. Baris 33 menjelaskan untuk mengatur data input pada kolom output (ID Kurir)
16. Baris 34-40 menjelaskan untuk mengatur data input pada kolom output (Nama Kurir)

5.4 Proses Inisialisasi Populasi Awal

Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan Algoritme Genetika secara umum dari inisialisasi data yang telah diinputkan. Kode program untuk implementasi fungsi inisialisasi data terdapat pada Kode Program 5.6.

Algoritma 6: Fungsi Inisialisasi Data	
1	<code>void loadJarak()</code>
2	<code>{</code>
3	<code>try</code>
4	<code>{</code>
5	<code>String q="SELECT * "+</code>
6	<code>"FROM pengirim p";</code>
7	<code>System.out.println(q);</code>
8	<code>PreparedStatement stmt =Gbaru.con.prepareStatement(q);</code>
9	<code>//stmt.setString(1, etUsername.getText());</code>
10	<code>//stmt.setString(2, etPassword.getText().toString());</code>
11	<code>ResultSet rs=stmt.executeQuery();</code>
12	<code>daftarKurir=new ArrayList<String>();</code>
13	<code>daftarPengirim=new ArrayList<String>();</code>
14	<code>model=new DefaultTableModel();</code>
15	<code>//model = new DefaultTableModel();</code>
16	<code>//JTable1 = new JTable(model);</code>
17	<code>model.addColumn("");</code>


```
18      model.addColumn("JNE");
19      while(rs.next())
20      {
21          model.addColumn(rs.getString("nama_pengirim"));
22          /*
23          model.addRow(new Object[]
24          {rs.getString("id"),rs.getString("no_pengirim"),
25            rs.getString("nama_pengirim"),
26            rs.getString("alamat_pengirim")
27          });*/
28          daftarPengirim.add(rs.getString("nama_pengirim"));
29      }
30      rs.close();

31      q="SELECT * FROM kurir p ";
32      stmt =Gbaru.con.prepareStatement(q);
33      //stmt.setString(1, etUsername.getText());
34      //stmt.setString(2, etPassword.getText().toString());
35      rs=stmt.executeQuery();
36      while (rs.next())
37      {
38          daftarKurir.add(rs.getString("nama"));
39      }
40      rs.close();

41      Object[] temp=new Object[daftarPengirim.size()+2];
42      temp[0]="";
43      for (int i=0;i<daftarPengirim.size()+1;i++)
44      {
45          //temp[i+1]="1";
46      }

47      for (int i=0;i<daftarPengirim.size()+1;i++)
48      {
49          for (int j=0;j<temp.length;j++)
50          {
51              temp[j]="";
52              //temp[i+1]="1";
53          }
54          if (i==0)
55          {
56              temp[0]="JNE";
57          }
58          else {
59              temp[0]=daftarPengirim.get(i-1);
60          }
61      }
```

62	<code>temp[i+1]=1;</code>
63	<code>model.addRow(temp);</code>
64	
65	<code>}</code>

Kode Program 5.6 Inisialisasi Data Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.6 adalah sebagai berikut:

1. Baris 1-11 menjelaskan parameter untuk menampilkan inisialisasi data
2. Baris 12 menjelaskan daftar kurir berbentuk *arraylist*
3. Baris 13 menjelaskan daftar pengirim berbentuk *arraylist*
4. Baris 14-17 menjelaskan membuat tabel baru
5. Baris 18 menjelaskan membuat daftar urutan rute diawali dari JNE (Kantor JNE)
6. Baris 19-30 menjelaskan untuk memanggil nama pengirim dari tabel daftar pengirim
7. Baris 31-35 menjelaskan pemilihan nama kurir
8. Baris 36-40 menjelaskan untuk memanggil nama kurir dari tabel daftar kurir
9. Baris 41-46 menjelaskan menambahkan nama pengirim pada daftar urutan rute
10. Baris 47-65 menjelaskan menambahkan nama pengirim pada daftar urutan rute hingga semua nama pengirim terfapat pada daftar urutan rute

5.5 Proses Reproduksi

Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan Algoritme Genetika secara umum dari reproduksi. Dimana proses reproduksi ini terdiri dari 2 proses yaitu *crossover* dan mutasi.

5.5.1 Proses Crossover

Kode program untuk implementasi fungsi *crossover* terdapat pada Kode Program 5.7.

Algoritma 7: Fungsi Crossover	
1	<code>public void tableChanged(TableModelEvent e) {</code>
2	<code> //if (equals(e).)</code>
3	<code> if (!changed)</code>
4	<code> {</code>
5	<code> changed=true;</code>
6	<code> int baris=e.getFirstRow();</code>
7	<code> int kolom=e.getColumn();</code>


```

8      String value=model.getValueAt(baris, kolom)+" ";
9      if (kolom==0)
10     {
11         int idx=0;
12
13         if (baris==0 && !value.equals("JNE"))
14         {
15             JOptionPane.showMessageDialog(null, "Tidak boleh mengganti kolom
16             judul");
17             model.setValueAt("JNE",baris, 0);
18         }
19         else if (baris!=0 && !value.equals(daftarPengirim.get(baris-1))) {
20             JOptionPane.showMessageDialog(null, "Tidak boleh mengganti kolom
21             judul");
22             model.setValueAt(daftarPengirim.get(baris-1),baris, 0);
23         }
24
25         //model.setValueAt(baris, kolom, value);
26     }
27     else if (baris+1==kolom)
28     {
29         if (!value.equals("1"))
30         {
31             JOptionPane.showMessageDialog(null, "Jarak terhadap diri sendiri harus
32             1");
33             model.setValueAt("1",baris, kolom);
34         }
35     }
36     else {
37         int tbarisbaru=kolom-1;
38         int tkolombaru=baris+1;
39
40         model.setValueAt(value,tbarisbaru, tkolombaru);
41     }
42     changed=false;
43 }
44
45 //System.out.println("Column: " + e.getColumn() + " Row: " + e.getFirstRow());
46 }
47 });
48
49 for (int i=0;i<daftarPengirim.size()+1;i++)
50 {
51     for (int j=0;j<daftarPengirim.size()+1;j++)
52     {
53         if (i+1!=j && j!=0)
54         {

```

```

55      int tbarisbaru=j-1;
56      int tkolombaru=i+1;
57      int nl=rdn.nextInt(10)+2;
58      model.setValueAt(nl,i,j);
59      model.setValueAt(nl,tbarisbaru,tkolombaru);
60  }
61  }
62  }
63  }
64  catch (Exception ex)
65  {
        ex.printStackTrace();
    }
}

```

Kode Program 5.7 Crossover Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.7 adalah sebagai berikut:

8. Baris 1 menjelaskan nama kelas untuk proses *crossover*
9. Baris 2-4 menjelaskan tentang pengujian apakah terjadi perubahan
10. Baris 5 menjelaskan apabila perubahan == benar;
11. Baris 6 menjelaskan parameter baris yang akan diubah
12. Baris 7 menjelaskan parameter kolom yang akan diubah
13. Baris 8 menjelaskan parameter mendapatkan nilai dari baris dan kolom yang telah dipilih
14. Baris 9-25 menjelaskan tentang pengujian apabila kolom ==0; memilih baris dan kolom yang akan diubah
15. Baris 26-27 menjelaskan tentang pengujian apabila baris+1==kolom;
16. Baris 28-29 menjelaskan tentang pengujian apabila nilai tidak sama dengan 1;
17. Baris 30-33 menjelaskan tentang menampilkan kotak dialog (Jarak Terhadap diri sendiri harus 1)
18. Baris 34-39 menjelaskan mengubah nilai baris baru dan kolom baru
19. Baris 40-44 menjelaskan else dari baris 5 apabila tidak ada perubahan == salah.
20. Baris 45-65 menjelaskan pengulangan untuk menampilkan hasil perubahan data pada kolom *output*

5.5.2 Proses Mutasi

Kode program untuk implementasi fungsi Mutasi terdapat pada Kode Program 5.8.

Algoritma 8: Fungsi Mutasi

```

1  public void tableChanged(TableModelEvent e) {
2      //if (equals(e).)
3      if (!changed)
4      {
5          changed=true;
6          int baris=e.getFirstRow();
7          int kolom=e.getColumn();
8          String value=model.getValueAt(baris, kolom)+"";
9          if (kolom==0)
10         {
11             int idx=0;
12
13
14             if (baris==0 && !value.equals("JNE"))
15             {
16                 JOptionPane.showMessageDialog(null, "Tidak boleh mengganti kolom
17                 judul");
18                 model.setValueAt("JNE",baris, 0);
19             }
20             else if (baris!=0 && !value.equals(daftarPengirim.get(baris-1))) {
21                 JOptionPane.showMessageDialog(null, "Tidak boleh mengganti kolom
22                 judul");
23                 model.setValueAt(daftarPengirim.get(baris-1),baris, 0);
24             }
25
26             //model.setValueAt(baris, kolom, value);
27         }
28         else if (baris+1==kolom)
29         {
30             if (!value.equals("1"))
31             {
32                 JOptionPane.showMessageDialog(null, "Jarak terhadap diri sendiri harus
33                 1");
34                 model.setValueAt("1",baris, kolom);
35             }
36         }
37         else {
38             int tbarisbaru=kolom-1;
39             int tkolombaru=baris+1;
40             model.setValueAt(value,tbarisbaru, tkolombaru);
41         }
42     }
43     changed=false;

```

```

44 }
45 }
46 //System.out.println("Column: " + e.getColumn() + " Row: " + e.getFirstRow());
    }
}
};

```

Kode Program 5.8 Mutasi Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.8 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk proses mutasi
2. Baris 2-4 menjelaskan tentang pengujian apakah terjadi perubahan
3. Baris 5 menjelaskan apabila perubahan == benar;
4. Baris 6 menjelaskan parameter baris yang akan diubah
5. Baris 7 menjelaskan parameter kolom yang akan diubah
6. Baris 8 menjelaskan parameter mendapatkan nilai dari baris dan kolom yang telah dipilih
7. Baris 9-25 menjelaskan tentang pengujian apabila kolom ==0; memilih baris dan kolom yang akan diubah
8. Baris 26-27 menjelaskan tentang pengujian apabila baris+1==kolom;
9. Baris 28-29 menjelaskan tentang pengujian apabila nilai tidak sama dengan 1;
10. Baris 30-33 menjelaskan tentang menampilkan kotak dialog (Jarak Terhadap diri sendiri harus 1)
11. Baris 34-39 menjelaskan mengubah nilai baris baru dan kolom baru
12. Baris 40-46 menjelaskan *else* dari baris 5 apabila tidak ada perubahan == salah;

5.6 Perhitungan Total Jarak Dan Fitness

Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan Algoritme Genetika secara umum dari Total Jarak dan *Fitness*. Kode program untuk implementasi fungsi Total Jarak dan *Fitness* terdapat pada Kode Program 5.9.

Algoritma 9: Fungsi Total Jarak dan *Fitness*

```

1 public double jarakTempuh(Calon cl)
2 {
3     int totalJarakTempuh=0;
4     int start=0;

```



```

5      for (int i=0;i<FormProses.daftarKurir.size();i++)
6      {
7          //hasil=hasil+FormProses.daftarKurir.get(i+":\n";
8
9          int jumlah=cl.jumlah.get(i);
10
11         String kirim="";
12         int i1=0;
13         int totalJarak=0;
14         for (int x=start;x<(start+jumlah);x++)
15         {
16             /*
17             if (!kirim.equals(""))
18             {
19                 kirim=kirim+"-";
20             }*/
21             int idx=Integer.parseInt(cl.calon.get(x));
22
23             int jarak=Integer.parseInt(FormProses.model.getValueAt(i1,idx+1)+"");
24             totalJarak+=jarak;
25             i1=idx;
26             //kirim=kirim+FormProses.daftarPengirim.get(idx);
27         }
28         //hasil=hasil+kirim+"\n";
29         //hasil=hasil+"Total Jarak "+totalJarak+"\n";
30         //hasil=hasil+"\n";
31         start=start+jumlah;
32
33         totalJarakTempuh+=totalJarak;
34
35     }
36     return totalJarakTempuh*1.0;
37
38 }

```

Kode Program 5.9 Total Jarak Dan *Fitness* Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.9 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk mengetahui jarak dan *fitness*
2. Baris 2-4 menjelaskan inisialisasi data
3. Baris 5-13 menjelaskan perulangan untuk mengetahui banyak jarak dari daftar kurir yang diproses
4. Baris 14-27 menjelaskan perulangan untuk mengetahui nilai jarak dari daftar kurir yang diproses
5. Baris 28-35 menjelaskan untuk menghitung total jarak tempuh

6. Baris 36-38 menjelaskan untuk menghitung nilai *fitness*

5.7 Proses Evaluasi Dan Seleksi

Proses Pada bagian proses implementasi ini merupakan penjelasan kode program dalam menjalankan Algoritme Genetika secara umum dari Evaluasi dan Seleksi. Kode program untuk implementasi fungsi evaluasi dan seleksi terdapat pada Kode Program 5.10.

Algoritma 10: Fungsi Evaluasi dan Seleksi

```

1  private void button1ActionPerformed(java.awt.event.ActionEvent evt) {
2  // TODO add your handling code here:
3  try
4  {
5      population=new ArrayList<String>();
6
7
8      dcalon=new ArrayList<Calon>();
9      int epoch=totalIterasi;
10     int totalPop=jumlahPopulasi;
11
12     double div=0;
13     for (int i=0;i<totalPop;i++)
14     {
15         ArrayList<String> calon=new ArrayList<>();
16         for (int x=0;x<daftarPengirim.size();x++)
17         {
18             calon.add(x+"");
19         }
20         for (int x=0;x<20;x++)
21         {
22             int i1=rdn.nextInt(calon.size());
23             int i2=rdn.nextInt(calon.size());
24
25             String calon1=calon.get(i1);
26             String calon2=calon.get(i2);
27
28             calon.set(i1,calon2);
29             calon.set(i2,calon1);
30         }
31
32         ArrayList<Integer> jumlahMasing=new ArrayList<Integer>();
33
34         //int jumlah1=(int)Math.round(rdn.nextInt((int)(calon.size()*0.8)));
35         //int jumlah2=(int)Math.round(rdn.nextInt((int)((calon.size()-jumlah1)*0.8)));
36         //int jumlah3=calon.size()-jumlah1-jumlah2;
37         int jumlahAwal=calon.size();

```



```

38     for (int x=0;x<daftarKurir.size()-1;x++)
39     {
40         int jumlah1=(int)Math.round(rdn.nextInt((int)(jumlahAwal*0.6)))+1;
41
42         jumlahMasing.add(jumlah1);
43         jumlahAwal=jumlah1;
44     }
45     jumlahMasing.add(jumlahAwal);
46
47     for (int x=0;x<20;x++)
48     {
49         int i1=rdn.nextInt(jumlahMasing.size());
50         int i2=rdn.nextInt(jumlahMasing.size());
51
52         int calon1=jumlahMasing.get(i1);
53         int calon2=jumlahMasing.get(i2);
54
55         jumlahMasing.set(i1,calon2);
56         jumlahMasing.set(i2,calon1);
57     }
58
59     Calon cl=new Calon();
60     cl.calon=calon;
61     cl.jumlah=jumlahMasing;
62     cl.jarakTempuh=jarakTempuh(cl);
63     div+=cl.jarakTempuh;
64     dcalon.add(cl);
65
66     //System.out.println("aa");
67
68     }
69
70
71     for (int i=0;i<dcalon.size();i++)
72     {
73         dcalon.get(i).jarakTempuh=dcalon.get(i).jarakTempuh/div;
74     }
75
76
77     sorting();
78     for (int i=0;i<epoch;i++)
79     {
80         sorting();
81         Calon baru=getBaru();
82         if (dcalon.get(dcalon.size()-1).jarakTempuh>baru.jarakTempuh)
83         {
84             dcalon.set(dcalon.size()-1,baru);
85         }
86     }

```

```

87
88 if (pn!=null)
89 {
90     pn.showDetail(dcalon.get(0));
91     setVisible(false);
92 }
93 else if (pn2!=null)
94 {
95     pn2.showDetail(dcalon);
96     setVisible(false);
97 }
98 }
99 }
100 catch (Exception ex)
101 {
102     ex.printStackTrace();
103     System.out.println("abc");
104 }
105 }

```

Kode Program 5.10 Evaluasi Dan Seleksi Algoritme Genetika

Untuk penjelasan pada implementasi kode program 5.10 adalah sebagai berikut:

1. Baris 1 menjelaskan nama kelas untuk memulai proses evaluasi dan seleksi
2. Baris 2-12 menjelaskan proses inisialisasi
3. Baris 13-15 menjelaskan perulangan untuk menampilkan *totalpop*
4. Baris 16-19 menjelaskan perulangan untuk menambahkan daftar pengirim
5. Baris 20-30 menjelaskan perulangan untuk mengetahui *size* dari calon, dan mendapatkan daftar calon
6. Baris 32-37 menjelaskan untuk mengatur jumlah awal == calon.size;
7. Baris 38-44 menjelaskan perulangan untuk menambahkan daftar kurir
8. Baris 45 menjelaskan menambahkan jumlah masing (jumlah awal)
9. Baris 47-57 menjelaskan perulangan untuk mengetahui *size* dari jumlah masing, dan mendapatkan daftar dari jumlah masing
10. Baris 59-68 menjelaskan inisialisasi data
11. Baris 71-74 menjelaskan dapatkan nilai jaraktempuh
12. Baris 75-105 menjelaskan dilakukannya proses *sorting* berdasarkan dari nilai *fitness* terbesar hingga terkecil

5.8 Implementasi Perancangan Antarmuka

Pada tampilan sistem akan dijelaskan hasil tampilan dari sistem yang telah diimplementasikan dan berjalan pada *platform desktop*. Tampilan sistem terdapat 3 halaman yaitu tampilan halaman Algoritma Genetika, halaman data, dan halaman Parameter.

5.8.1 Tampilan Halaman Data

Pada bagian halaman data akan menunjukkan data yang sudah dimasukkan pada proses *input* pada halaman *input*. Terdapat data pengirim yang berisi data no pengirim, nama pengirim dan alamat pengirim. Dan terdapat data kurir yang berisi data no ID dan nama kurir. Hasil tampilan halaman data terdapat pada Gambar 5.1

Optimasi Pengiriman Barang Menggunakan Algoritma Genetika

Data Parameter Algoritma Genetika

Data Pengirim

No Pengirim	Nama Pengirim	Alamat Pengirim

Data Kurir

ID	Nama

Gambar 5.1 Hasil Tampilan Data

5.8.2 Tampilan Halaman Parameter

Pada bagian inisialisasi data akan menampilkan dua input data yang meliputi data pengirim barang dan data kurir. Terdapat menu utama yaitu menu halaman Data, halaman Parameter, halaman Algoritma Genetika. Terdapat tombol *add* pengirim dan *add* kurir untuk menambah data pengirim dan kurir dimana sebelumnya mengisi kolom – kolom isian data pengirim dan kurir. Juga terdapat tombol proses untuk melakukan proses data yang telah diinputkan. Dan kolom untuk menampilkan

data yang baru saja dimasukkan. Hasil tampilan halaman *input* terdapat pada Gambar 5.2.

Optimasi Pengiriman Barang Menggunakan Algoritma Genetika

Data Parameter Algoritma Genetika

Input Pengirim

No Pengirim

Nama Pengirim

Alamat Pengirim

Add Pengirim

Input Kurir

No ID

Nama Kurir

Proses

Add Kurir

Gambar 5.2 Hasil Tampilan Halaman Parameter

5.8.3 Tampilan Halaman Algoritme Genetika

Pada bagian halaman Algoritme Genetika adalah halaman untuk menampilkan hasil *output* data dari hasil proses algoritme genetika. Terdapat kolom *output* untuk menampilkan hasil rute yang didapatkan dari hasil algoritme genetika. Hasil tampilan halaman *output* terdapat pada Gambar 5.3.

Optimasi Pengiriman Barang Menggunakan Algoritma Genetika

Data Parameter Algoritma Genetika

CR

MR

Jumlah Pop Size

Jumlah Gen

Set

Output

Urutan Pengiriman	Kurir	Total Jarak	Nilai Fitness	Generasi Ke

Gambar 5.3 Hasil Tampilan Halaman Algoritme Genetika

BAB 6 HASIL DAN PEMBAHASAN

Hasil pengujian berdasarkan nilai *fitness* yang diperoleh dari implementasi sistem. Hasil maupun pembahasan terkait hasil pengujian yang akan dilakukan adalah pertama uji coba ukuran populasi dan pembahasan hasil, kedua uji coba banyak generasi dan pembahasan hasil, ketiga uji coba kombinasi Cr & Mr dan pembahasan hasil.

6.1 Uji Coba Ukuran Populasi Dan Pembahasan Hasil

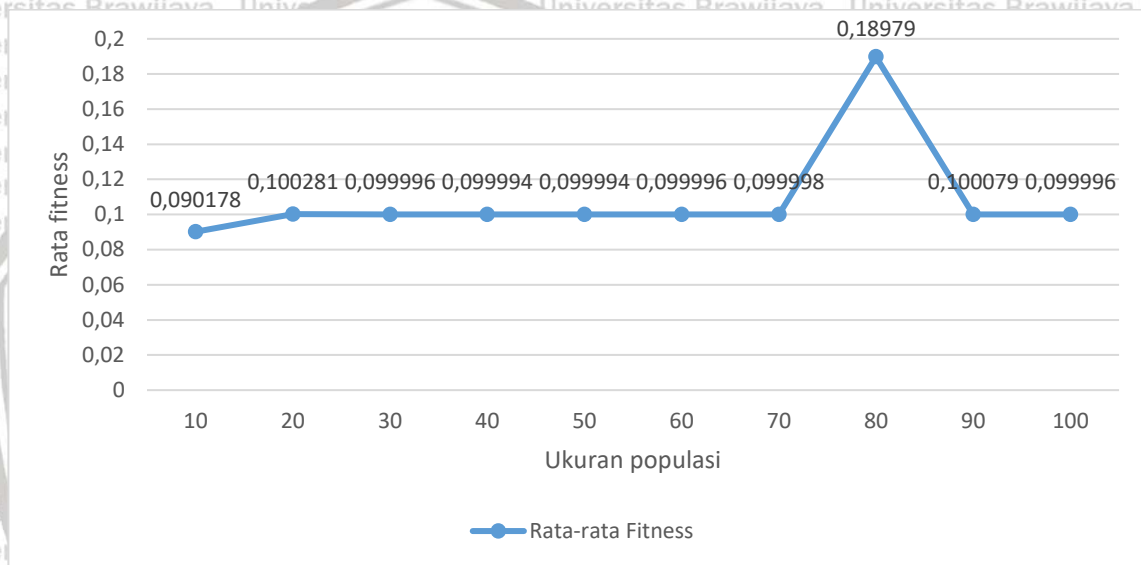
Pada bagian pengujian ukuran populasi ini dilakukan untuk mengetahui ukuran populasi pada setiap percobaan yang telah dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan. Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap-tiap percobaan yang didapatkan per ukuran populasi. Dan juga didapatkan nilai rata-rata *fitness* dari semua nilai percobaan yang didapatkan pada per ukuran populasi. Proses uji coba dilakukan dengan melakukan pengujian sebanyak 10 kali percobaan dengan menggunakan kelipatan 10 hingga 100 dari ukuran populasi. Proses pengujian ini menggunakan parameter dengan banyak generasi 10, *crossover rate* 1,0 dan *mutation rate* 0,0. Banyak Nama pengirim 25 dan banyak kurir 3. Uji coba ukuran populasi dapat dilihat pada Tabel 6.1 dan hasil secara detail terdapat pada Lampiran D.

Tabel 6.1 Uji Coba Ukuran Populasi

Ukuran populasi	Nilai Fitness Percobaan ke-n						Fitness tertinggi	Rata-rata fitness
	1	2	3	...	9	10		
10	0,08753	0,08885	0,09681	...	0,10610	0,11405	0,11405	0,090178
20	0,09012	0,09227	0,09298	...	0,10515	0,11731	0,11731	0,100281
30	0,08061	0,09183	0,09336	...	0,11021	0,11581	0,11581	0,099996
40	0,09171	0,09317	0,09647	...	0,10344	0,11555	0,11555	0,099994
50	0,09247	0,09803	0,09803	...	0,10359	0,10418	0,10418	0,099994
60	0,09286	0,09669	0,09746	...	0,10591	0,10872	0,10872	0,099996
70	0,09094	0,09367	0,09786	...	0,10331	0,10938	0,10938	0,099998
80	0,09559	0,09559	0,09711	...	0,10603	0,10735	0,10735	0,18979

90	0,09641	0,09792	0,09792	...	0,10332	0,10467	0,10467	0,100079
100	0,09381	0,09758	0,09899	...	0,10213	0,10276	0,10276	0,099996

Dari hasil Tabel 6.1 diperoleh hasil rata-rata nilai fitness yang didapatkan dari 10 kali percobaan pada uji coba ukuran populasi yang ditampilkan dalam diagram garis. Berikut ini merupakan diagram garis untuk nilai rata-rata fitness hasil uji coba ukuran populasi terdapat pada bagian Gambar 6.1:



Gambar 6.1 Hasil Uji Coba Ukuran Populasi

Hasil pengujian pada bagian Tabel 6.1 dan hasil grafik uji coba pada Gambar 6.1 menunjukkan nilai ukuran populasi pada nilai *fitness*. Dari gambar diatas, dapat diperoleh nilai rata-rata *fitness* paling tinggi yang terdapat pada ukuran populasi 80 dengan hasil rata-rata *fitnes* hampir mendekati 2 yaitu 0,18979. Hal ini membuktikan bahwa ukuran populasi menentukan besar kecilnya rata-rata nilai *fitness*.

6.2 Uji Coba Banyak Generasi Dan Pembahasan Hasil

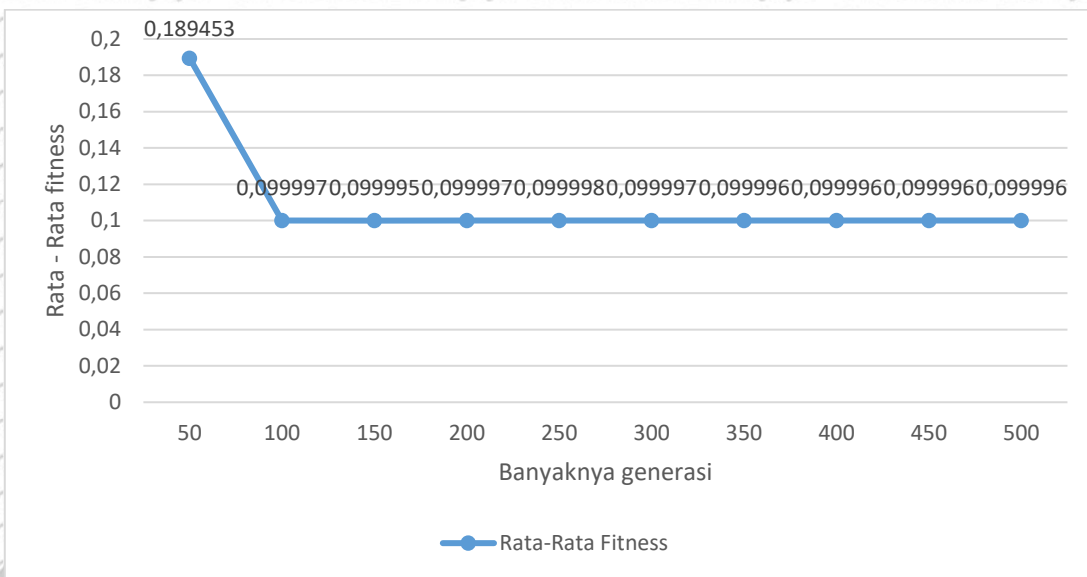
Pada bagian pengujian banyak generasi ini dilakukan untuk mengetahui banyak generasi pada setiap percobaan yang telah dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan. Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap-tiap percobaan yang didapatkan per banyak generasi. Dan juga didapatkan nilai rata-rata *fitness* dari semua nilai percobaan yang didapatkan pada per banyak generasi.

Proses uji coba dilakukan dengan melakukan pengujian sebanyak 10 kali percobaan yang menggunakan kelipatan 50 hingga 500 dari ukuran populasi. Proses pengujian ini menggunakan parameter dengan *popsi* 10, *crossover rate* 1,0 dan *mutation rate* 0,0. Banyak Nama pengirim 25 dan banyak kurir 3. Perancangan uji coba banyak generasi dapat dilihat pada Tabel 6.2 dan hasil secara detail terdapat pada Lampiran E.

Tabel 6.2 Uji Coba Banyak Generasi

Banyak Generasi	Nilai Fitness Percobaan ke - n						Fitness tertinggi	Rata – rata fitness
	1	2	3	...	9	10		
50	0,09546	0,09607	0,09728	...	0,10438	0,10725	0,10725	0.189453
100	0,09248	0,09478	0,09555	...	0,10691	0,10812	0,10812	0.099997
150	0,09523	0,09774	0,09853	...	0,10325	0,10373	0,10373	0.099995
200	0,09386	0,09751	0,09781	...	0,10343	0,10586	0,10586	0.099997
250	0,09498	0,09789	0,09805	...	0,10372	0,10541	0,10541	0.099998
300	0,09565	0,09611	0,09751	...	0,10357	0,10451	0,10451	0.099997
350	0,09447	0,09602	0,09664	...	0,10361	0,10437	0,10437	0.099996
400	0,09737	0,09807	0,09891	...	0,10158	0,10214	0,10214	0.099996
450	0,08974	0,09303	0,09350	...	0,10493	0,10806	0,10806	0.099996
500	0,09202	0,09564	0,09815	...	0,10429	0,10744	0,10744	0.099996

Dari hasil Tabel 6.2 diperoleh hasil nilai fitness sebanyak 10 percobaan pada uji coba banyaknya generasi yang disajikan dalam bentuk diagram garis. Berikut diagram garis untuk hasil uji coba banyaknya generasi terdapat pada bagian Gambar 6.2.



Gambar 6.2 Hasil Uji Coba Banyaknya Generasi

Berdasarkan dari hasil pengujian pada bagian Tabel 6.2 dan hasil grafik uji coba pada Gambar 6.2 menunjukkan nilai banyaknya generasi terhadap rata-rata nilai *fitness* yang didapatkan. Nilai *fitness* rata-rata yang tertinggi terjadi pada ukuran generasi 50 dengan hasil rata – rata *fitness* hampir mendekati 2 yaitu 0,189453.

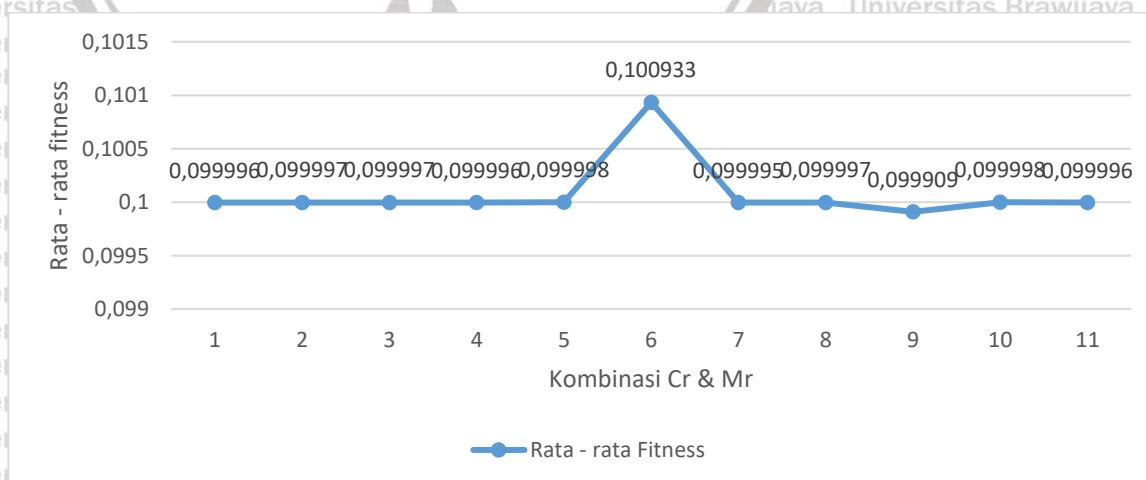
6.3 Uji Coba Kombinasi Cr & Mr Dan Pembahasan Hasil

Pada bagian kombinasi nilai Cr & Mr ini dilakukan untuk mengetahui kombinasi nilai Cr & Mr pada setiap percobaan yang telah dilakukan. Dimana setiap kali dilakukan percobaan maka akan menghasilkan nilai *fitness* pada tiap percobaan yang dilakukan. Dari hasil tersebut maka akan didapatkan nilai *fitness* tertinggi dari tiap – tiap percobaan yang didapatkan per kombinasi nilai Cr & Mr. Dan juga didapatkan nilai rata – rata *fitness* dari semua nilai percobaan yang didapatkan pada per kombinasi nilai Cr & Mr. Proses uji coba dilakukan dengan melakukan pengujian kombinasi nilai Cr & Mr sebanyak 10 kali percobaan dengan menggunakan Cr kelipatan 0,1 dimulai 0 hingga 1 dan menggunakan Mr kelipatan 0,1 menurun dimulai dari 1 hingga 0. Pada setiap ukuran populasi dilakukan sebanyak 10 kali percobaan. Proses pengujian ini menggunakan parameter sebanyak generasi 20, *popsiz* 10. Banyak Nama pengirim 25 dan banyak kurir 3. Perancangan uji coba kombinasi nilai Cr & Mr dapat dilihat pada Tabel 5.3 dan hasil secara detail terdapat pada Lampiran

Tabel 6.3 Uji Coba Kombinasi Cr & Mr

Kombinasi		Nilai Fitness Percobaan ke-n					Fitness tertinggi	Rata – rata fitness
Cr	Mr	1	2	...	9	10		
0	1	0,09391	0,09557	...	0,10496	0,10496	0,10496	0,099996
0.1	0.9	0,09597	0,09663	...	0,10317	0,10398	0,10398	0,099997
0.2	0.8	0,09281	0,09541	...	0,10231	0,10691	0,10691	0,099997
0.3	0.7	0,09673	0,09688	...	0,10474	0,10504	0,10504	0,099996
0.4	0.6	0,09428	0,09584	...	0,10471	0,10595	0,10595	0,099998
0.5	0.5	0,09157	0,09471	...	0,10564	0,10829	0,10829	0,100933
0.6	0.4	0,09247	0,09673	...	0,10272	0,10825	0,10825	0,099995
0.7	0.3	0,09579	0,09809	...	0,10175	0,10618	0,10618	0,099997
0.8	0.2	0,09557	0,09608	...	0,10329	0,10404	0,10404	0,099909
0.9	0.1	0,09338	0,09663	...	0,10357	0,10416	0,10416	0,099998
1	0	0,09581	0,09581	...	0,10335	0,10415	0,10415	0,099996

Dari hasil Tabel 6.3 diperoleh hasil nilai *fitness* sebanyak 10 percobaan pada uji coba kombinasi Cr & Mr yang disajikan dalam bentuk diagram garis. Berikut diagram garis untuk nilai *fitness* dari hasil uji coba kombinasi Cr & Mr terdapat pada bagian Gambar 6.3.



Gambar 6.3 Hasil Uji Coba Kombinasi Cr & Mr

Berdasarkan hasil pengujian pada bagian Tabel 6.3 dan hasil grafik uji coba pada Gambar 6.3 menunjukkan nilai kombinasi Cr & Mr terhadap rata – rata *fitness*. Rata-rata *fitness* tertinggi pada kombinasi ke 6 yaitu Crossover rate = 0,5 & Mutation rate = 0,5. Dengan hasil nilai rata – rata *fitness* yaitu 0.100933.

6.2 Analisis Global

Pada bagian pengujian Sistem ini dilakukan dengan melakukan perbandingan hasil perhitungan manualisasi dengan hasil perhitungan dari sistem. Dimana hasil perhitungan manualisasi merupakan hasil pengiriman yang dilakukan oleh JNE tanpa bantuan sistem. Untuk uji coba sistem ini data yang digunakan yaitu urutan rute yang dilakukan saat pengiriman barang pada JNE dengan beberapa kurir yang melakukan kunjungan ke beberapa alamat yang dituju. Pengujian perhitungan manualisasi dapat dilihat pada Tabel 6.4.

Tabel 6.4 Uji Coba Perhitungan Hasil Pengiriman Manualisasi

Nama Kurir	Kunjungan	Total Jarak
Andreas Mangan	ZAN – RU– AM– RA– MA– ZNR– AE– YS– UA– RP– DF– PA– RS– AF– SH– GA– HP– CP– MU– IP– YS– LAP– RP– IR– NNI– EP– JS	223
Arbi Bisalim	MA – AF– AE– SH– UA– NNI– DF– PA– RS– ZAN– GA– RP– CP– MU– IP– AK– MS– RS– MW	160
Edo Farah	PA– RP– NNI– SH– GA– HP– CP– MU– IP– YS– LAP– AR– IR– AW– ZAN– RU– DD– RA– MA– ZNR– AM– AE– YS– UA– RP– DF– DR– FA– R– A– DA– DW– AP– DP– GA– D– CA– AP– AT	286
	Fitness	0,100360036

Berdasarkan tabel 6.4 diketahui hasil yang didapatkan dalam proses perhitungan manualisasi untuk urutan rute pengiriman barang pada JNE dengan total jarak yang dilalui pada masing-masing kurir. Pada kurir pertama yaitu Andreas Mangan yang ditugaskan untuk mengirim barang dengan total jarak 223 km, Arbi Bisalim yang mendapat tugas untuk mengirim barang dengan total jarak 160 km, Edo Farah juga ditugaskan untuk mengirim barang dengan total jarak 286 km. Selanjutnya pada pengujian Sistem ini dilakukan berdasarkan urutan rute yang memiliki nilai *fitness* terbaik. Terbaik dalam hal ini adalah nilai *fitness* tertinggi dari hasil uji coba yang telah dilakukan diatas. Perancangan uji coba Sistem dapat dilihat pada Tabel 6.5.

Tabel 6.5 Uji Coba Rekomendasi Sistem

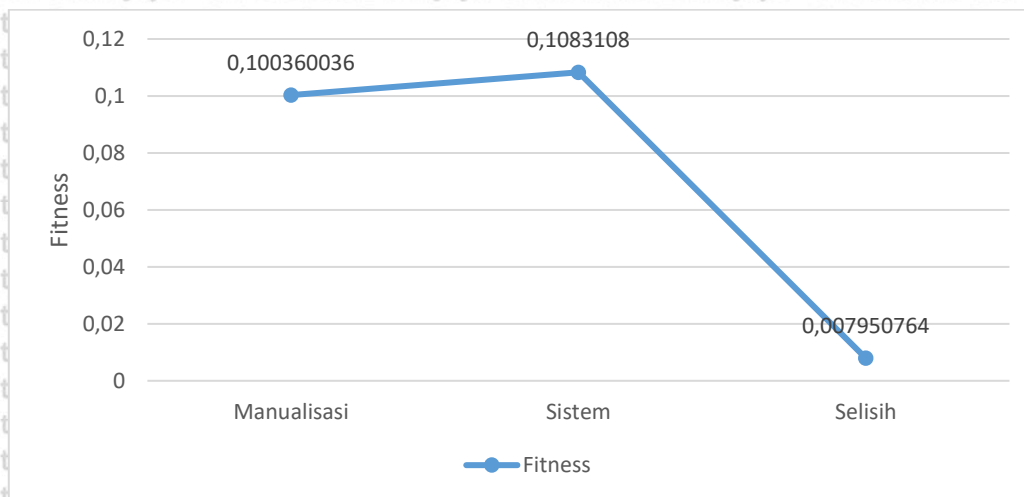
Nama Kurir	Kunjungan	Total Jarak
Andreas Mangan	AW-JS-JA-MS-RT-FW-RU-RA-AM-MA-ZNR-AE-HP-FA-RP-AR-ZAN-R-A-JSu-UF-GA-AP-WA-DA-CP-MU-IP-YS-YSu-N-DP-GA-DD	234
Arbi Bisalim	RP-EP-LAP-AT-WH-D-FA-SM-TR-AF-MB-FA-MS-DW-DF-K-DR-YF-PAm-Fap-Km-RS	152
Edo Farah	YS-AY-AYa-AH-FD-DA-NNI-WD-PA-CA-JS-AK-AG-RS-R-AP-FD-EW-D-KK-MW-AR-AF-RR-RT-PD-KS-RS-HS-NR-DA-DF-DD-LF-DDA-AH-ND-IC-AA-SF-A-SH-SJ-FB-IR	336
	Fitness	0,1083108

Berdasarkan tabel 6.5 diketahui hasil yang diperoleh dalam proses perhitungan menggunakan aplikasi untuk urutan rute pengiriman barang pada JNE dengan total jarak yang dilalui pada masing-masing kurir. Pada kurir pertama yaitu Andreas Mangan mendapat tugas mengirim barang dengan total jarak 234 km, Arbi Bisalim mendapat tugas mengirim barang dengan total jarak 152 km, Edo Farah mendapat tugas mengirim barang dengan total jarak 336 km. Terakhir, uji coba yang dilakukan adalah untuk mengetahui hasil perbandingan rekomendasi sistem dan manualisasi. Perancangan uji coba hasil perbandingan rekomendasi sistem dan manualisasi dapat dilihat pada Tabel 6.6.

Tabel 6.6 Uji Coba Perbandingan Rekomendasi Sistem dan Manualisasi

Kurir	Kurir 1 (KM)	Kurir 2 (KM)	Kurir 3 (KM)	Total jarak	Fitness
Hasil pengiriman JNE	223	160	286	669	0,100360036
Hasil Rekomendasi sistem	234	152	336	722	0,1083108
Selisih	11	8	50	53	0,007950764

Dari hasil Tabel 6.6 diperoleh hasil nilai *fitness* yang dihasilkan dari perbandingan sistem rekomendasi dan manualisasi yang ditampilkan dalam bentuk diagram garis. Berikut diagram garis untuk hasil perbandingan sistem rekomendasi dan manualisasi terdapat pada bagian Gambar 6.4.



Gambar 6.4 Hasil Uji Coba Perbandingan Sistem Rekomendasi Dan Manualisasi

Berdasarkan hasil pengujian pada bagian Tabel 6.4, Tabel 6.5, Tabel 6.6 dan hasil grafik uji coba pada Gambar 6.4 menunjukkan hasil perbandingan sistem rekomendasi dan manualisasi terhadap rata-rata *fitness*. *Fitness* tertinggi terdapat pada Hasil Rekomendasi Sistem yang menghasilkan nilai *fitness* yaitu 0,1083108. Hal ini membuktikan bahwa hasil rekomendasi sistem yang didapatkan lebih baik daripada sistem manualisasi pengiriman JNE yang mendapatkan *fitness* sebesar 0,100360036. Selisih antara hasil manualisasi pengiriman JNE dengan hasil rekomendasi sistem adalah 0,007950764.

BAB 7 PENUTUP

Pada Bab penutup ini akan menjelaskan mengenai kesimpulan maupun saran bagi penelitian pada optimasi pengiriman barang menggunakan algoritma genetika. Pada bagian kesimpulan akan menjawab beberapa pertanyaan yang telah disebutkan pada rumusan masalah di penelitian ini. Penjelasan mengenai kekurangan yang ditemukan dalam penelitian ini akan disampaikan pada bagian saran sehingga dapat digunakan sebagai acuan dalam penelitian berikutnya.

7.1 Kesimpulan

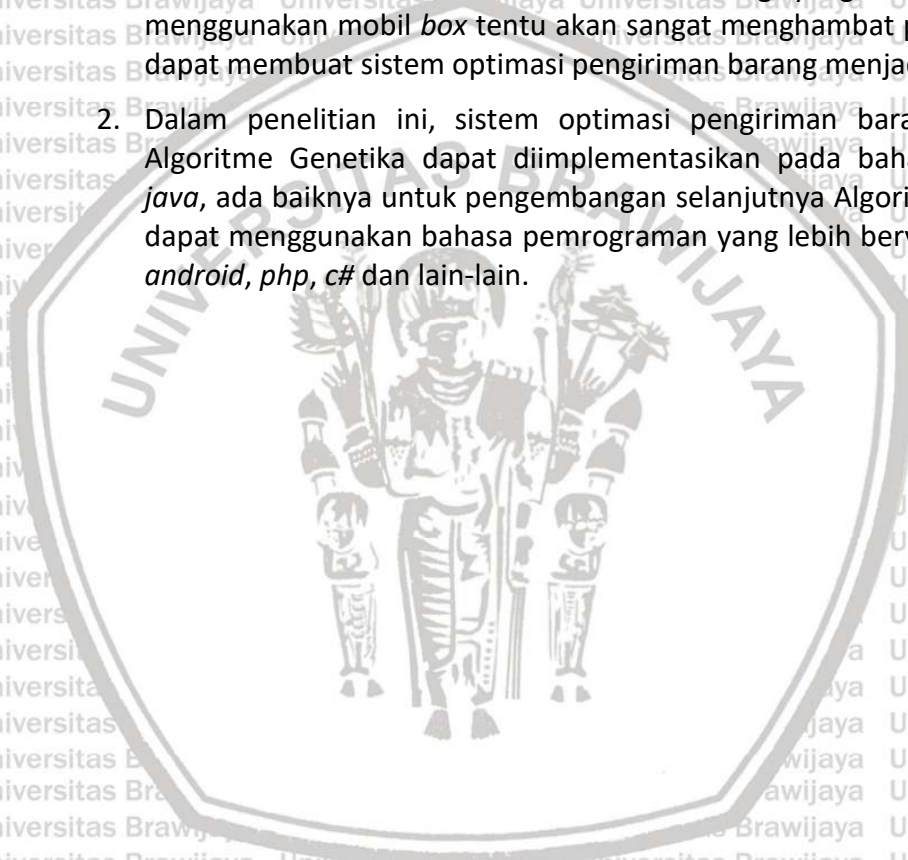
Berdasarkan dari hasil dan pembahasan pengujian pada Bab sebelumnya mengenai “Optimasi Pengiriman Barang Menggunakan Algoritme Genetika dengan Data Sintesis”. Kesimpulan yang didapatkan dari penelitian ini adalah sebagai berikut:

1. Algoritme Genetika untuk optimasi pengiriman barang menghasilkan nilai parameter yang optimal sebagai berikut:
 - Pengujian populasi ini dilakukan sebanyak 10 kali percobaan dengan banyaknya populasi yang berbeda-beda. Secara umum, jika ukuran populasi semakin banyak, maka nilai *fitness* yang diperoleh akan semakin bagus. Pada penelitian ini hasil ukuran populasi berdasarkan beberapa hasil pengujian yang optimal adalah menggunakan 80 populasi dengan nilai *fitness* 0,18979.
 - Pengujian banyak generasi juga dilakukan sebanyak 10 kali. Dari beberapa pengujian, didapatkan jumlah generasi yang paling optimal yaitu jumlah generasi 50 dengan hasil nilai *fitness* sebesar 0,189453.
 - Pengujian kombinasi *Cr* dan *Mr* ini juga dilakukan sebanyak 10 kali percobaan. Dari pengujian tersebut diperoleh hasil yang optimal yaitu nilai *Cr* sebesar 0,5, *Mr* sebesar 0,5. Dengan hasil nilai *fitness* yaitu 0,100933.
2. Algoritme Genetika untuk optimasi pengiriman barang menghasilkan tingkat keberhasilan yang optimal. Hal ini dibuktikan dari hasil analisis global yang telah dilakukan, dimana analisis global merupakan perbandingan dari hasil perhitungan manualisasi dengan hasil aktual sistem. Perhitungan manualisasi merupakan hasil pengiriman yang dilakukan oleh JNE tanpa bantuan sistem. Didapatkan selisih *fitness* antara hasil manualisasi pengiriman JNE dengan hasil rekomendasi sistem adalah 0,007950764. Dengan hasil rekomendasi sistem yang mendapatkan *fitness* 0,1083108 lebih baik daripada hasil manualisasi yang menghasilkan *fitness* sebesar 0,1003060036.

7.2 Saran

Saran yang digunakan untuk penelitian selanjutnya adalah sebagai berikut:

1. Penambahan parameter pengiriman yang digunakan yaitu perbedaan volume dan berat barang yang akan dikirimkan ke titik alamat penerima. Karena volume dan berat barang juga mempengaruhi kendaraan yang akan dibawa oleh kurir. Jika kurir menggunakan kendaraan motor dengan membawa beban volume dan berat barang yang melebihi kurir yang menggunakan mobil *box* tentu akan sangat menghambat pengiriman. Hal ini dapat membuat sistem optimasi pengiriman barang menjadi kurang optimal.
2. Dalam penelitian ini, sistem optimasi pengiriman barang menggunakan Algoritme Genetika dapat diimplementasikan pada bahasa pemrograman *java*, ada baiknya untuk pengembangan selanjutnya Algoritme Genetika juga dapat menggunakan bahasa pemrograman yang lebih bervariasi seperti *java android*, *php*, *c#* dan lain-lain.



DAFTAR REFERENSI

- Ahn, C. W. & Ramakrishna, R. S., (2002). A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, Volume 6.
- Arif, M. B. et al., (2019). Penentuan Jarak Terpendek Pada Jalur Pengiriman Musae Chips Dengan Menggunakan Algoritma Genetika. *Jurnal Teknologi Informasi dan Trapan (J-TIT)*, Volume 6, pp. 19-23.
- Baskara Joni, I. D. M. A. & Nurcahyani, V., (2012). Penentuan Jarak Terpendek Pada Jalur Distribusi Barang Di Pulau Jawa Dengan Menggunakan Algoritma Genetika. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, Volume 1, pp. 244-258.
- Basuki, M., (2017). Penentuan Rute Optimum Distribusi Produk PT Indmira Berdasarkan Jarak. *Jurnal Desminasi Teknologi*, Volume 5, pp. 1-7.
- Dr. Wahidmurni, M., (2017). PEMAPARAN METODE PENELITIAN KUENTITATIF. *Jurnal Fakultas Ilmu Tarbiyah dan Keguruan UIN Maulana Malik Ibrahim Malang*.
- Hanafi, A., Wihandika, R. C. & Adikara, P. P., (2018). Penyelesaian Multiple Travelling Salesman Problem (M-TSP) Dengan Menggunakan Algoritme Genetika: Studi Kasus Pendistribusian Barang Di Kantor Pos Lumajang. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 2000-2006.
- Hannawati, A., Thiang & Eleazar, (2002). Pencarian Rute Optimum Menggunakan Algoritma Gnetetika. *Teknik Elektro*, Volume 2, pp. 78-83.
- Hasibuan, M. D. A. C. & Lusiana, (2015). Pencarian Rute Terbaik Pada Travelling Salesman Problem (TSP) menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru. *SATIN - Sain dan Teknologi Informasi*, pp. 47-45.
- Hasyim, N. M., Djamal, E. C. & Komarudin, A., (2017). Optimalisasi Rute Obyek Wisata Di Bandung Raya Menggunakan Algoritma Genetika. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, pp. 6-10.
- Hutasoit, E. T. H., (2019). Pencarian Rute Terpendek Menggunakan Algoritma Bellman-Ford (Studi Kasus: PT. JNE Medan). *Jurnal Sistem Komputer dan Informatika(JSON)*, Volume 1, pp. 20-25.
- Kusrini & Istiyanto, J. E., (2007). Penyelesaian Travelling Salesman Problem Algoritma Cheapest Insertion Heuristics dan Basis Data. *Jurnal Informatika*, Volume 8, pp. 109-114.

- Laksono, A. T., Utami, M. C. & Sugiarti, Y., (2016). Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran dan Kesehatan Universitas Muhammadiyah Jakarta). *Jurnal Sistem Informasi*, pp. 177-188.
- Lestari, E., Sulistianingsih, E. & Imro'ah, N., (2019). Penentuan Portopolio Saham Optimal Menggunakan Algoritma Genetika. *Buletin Ilmiah Mat. Stat dan Terapannya (Bimaster)*, Volume 08, pp. 193-200.
- Lubis, N. & Ginting, G., (2016). Perancangan Aplikasi Penjadwalan Kereta Api dengan Menggunakan Algoritma Genetika (Studi Kasus : Stasiun Kereta Api Bandar Khalipah Medan). *Jurnal Riset Komputer (JURIKOM)*, Volume 3, pp. 110-114.
- Lukas, S., Anwar, T. & Yuliani, W., (2005). Penerapan Algoritma Genetika untuk Travelling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2005)*, pp. 1-5.
- Mahmudy, W. F., (2015). *Dasar-Dasar Algoritma Evolusi*. Universitas Brawijaya Malang: Program Teknologi Informasi dan Ilmu Komputer.
- Natalia, C., Setiadi, G. G. & Silalahi, A., (2019). Optimasi Pengiriman Sapi Potong pada Jaringan Pelayaran Kapal Khusus Ternak dengan Algoritma Genetika. *Jurnal Metris* 20, pp. 51-57.
- Permanasari, Y. & Salim, R. A., (2006). Representasi Jalur (PATH) Pada Travelling Salesman Problem untuk Menentukan Jarak Terpendek menggunakan Algoritma Genetika. *Jurnal Matematika*, Volume 6, pp. 55-62.
- Pratama, R. A., Djamal, E. C. & Komarudin, A., (2017). Optimalisasi Pengantaran Barang dalam Perdagangan Online Menggunakan Algoritma Genetika. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, pp. 11-15.
- Raditya, P. M. R. & Dewi, C., (2018). Optimasi Multiple Travelling Salesman Problem (M-TSP) Pada Penentuan Rute Optimal Penjemputan Penumpang Travel Menggunakan Algoritma Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 3560-3568.
- Rizki, A. M., Mahmudy, W. F. & Yuliastuti, G. E., (2017). Optimasi Multi Travelling Salesman Problem (M-TSP) Untuk Distribusi Produk Pada Home Industri Tekstil Dengan Algoritma Genetika. *Kumpulan Jurnal Ilmu Komputer (KLIK)*, Volume 04, pp. 125-135.
- Sahamer & Mahmudy, W. F., (2015). Penerapan Algoritma Genetika Untuk Memaksimalkan laba produksi Jilbab. *Journal of Environmental Engineering & Sustainable Technology (JEEST)*, Volume 02, pp. 06-11.

Saputro, H. A., Mahmudy, W. F. & Dewi, C., (2015). Implementasi Algoritma Genetika Untuk Optimalisasi Penggunaan Lahan Pertanian. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*.

Sari, F. A., Sugiharti, E. & D., (2013). Implementasi Algoritma Genetika Untuk Menyelesaikan Travelling Salesman Problem. *UNNES Journal of Mathematics*, pp. 116-120.

Sari, R. N. & Mahmudy, W. F., (2015). Penyelesaian Multiple Travelling Salesperson Problem (M-TSP) Dengan Algoritma Genetika : Studi Kasus Pendistribusian Air Mineral. *Jurnal Skripsi. DORO : Jurnal Mahasiswa PTIK Universitas Brawijaya*, Volume 5.

Sinaga, R. A., (2019). *Optimalisasi Rute Pengiriman Paket Pada Perusahaan Dengan Penerapan Algoritma Genetika*. Medan: Universitas Sumatera Utara.

Sofwan, A., Handoyo, E. & WD, R., (2008). Algoritma Genetika dalam Pemilihan Spesifikasi Komputer. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2008)*, pp. 1-6.

Sukarmawati, Y., N. & Hartono, D. M., (2013). Optimalisasi Rute Pengumpulan Sampah Di Kawasan Perumahan Pesona Khayangan Dengan Model Penyelesaian Travelling Salesman Problem. *Jurnal Transportasi*, Volume 13, pp. 1-8.

Suprayogi, D. A. & Mahmudy, W. F., (2015). Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry. *Jurnal Buana*, Volume 6, pp. 121-130.

Supriana, I. W., (2017). Implementasi Algoritma Genetika dalam Penentuan Rute Terbaik Pendistribusian BBM Pada PT Burung Laut. *Jurnal Teknologi Informasi dan Komputer*, Volume 3, pp. 285-294.

Supriati, R., Irmawan, A. Y. & Melandy, G., (2017). Sistem Informasi Pengiriman Surat dan Barang Pada PT. POS Indonesia Kota Tangerang. *Strategic of Education in Information System*, Volume 3, pp. 173-181.

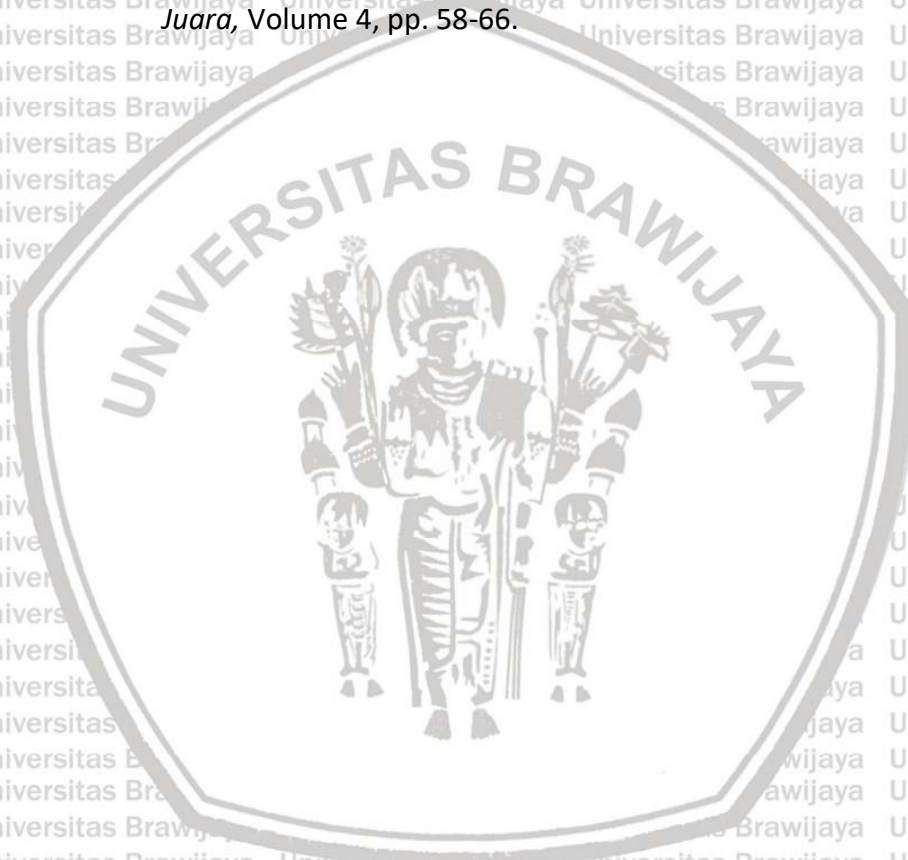
Surya Dharma, M. P., (2008). *Pendekatan, Jenis, Dan Metode Penelitian Pendidikan*. Kompetensi Penelitian dan Pengembangan 05 - B1 ed. Jakarta: Pengawas Sekolah Pendidikan Menengah.

Suwarnayanti, N. L. G. P., Sudarsana, I. M. & Darmayasa, S., (2016). Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran. *Journal of Applied Intelligent System*, Volume 1, pp. 220-233.

Tahyudin, I. & Susanti, I., (2015). Pencarian Rute Terbaik pada Obyek Wisata di Kabupaten Banyumas Menggunakan Algoritma Genetika Metode TSP (Travelling Salesman Problem). *JUITA*, Volume 3, pp. 165-173.

Triyani, E., (2017). *EVALUASI SISTEM PENGENDALIAN INTERN PENJUALAN KREDIT (Studi Kasus pada PT.ADEX)*. Yogyakarta: Universitas Mercu Buana.

Wibowo, H., Handayani, E. P., Wijaya, D. & Octaviani, T., (2019). Prosedur Pengiriman Barang pada PT POS Indonesia (PERSERO) Kantor POS Jakarta Selatan. *Akbar Juara*, Volume 4, pp. 58-66.



LAMPIRAN A DATA ALAMAT PENGIRIM

No	Nama	Alamat	Jarak dari kantor JNE
1	RU	Jl. Raya Delikguno	3.5
2	RA	Jl. Ki Ageng Sindujoyo	2.2
3	AM	Jl. Veteran	2.7
4	MA	Jl. Bangkal	6.2
5	ZNR	Jl. Sunan Drajat	2
6	AE	Jl. Semangka	1.7
7	HP	Jl. Melati	1.1
8	RS	Jl. Aknela	4.5
9	RP	Jl. Mlaten Gang 1	3.3
10	DF	Jl. Made Kidu Gang 22	5.1
11	ZAN	Jl. Rangge I	3
12	PA	Jl. Raya Pegaron	2.5
13	LAP	Jl. Raya Topeng	1.8
14	RPu	Jl. Kramat Raya	1.3
15	UA	Jl. Sumargo	2.3
16	SH	Jl. Raya Sari Rejo	6
17	GA	Jl. Raya Tambakboyo	4.6
18	IR	Jl. Ikan Lele Raya	3.6
19	AF	Jl. Andansari III	2.9
20	CP	Jl. Wajik	0.8
21	MU	Jl. Rancang Kencono	7.4
22	IP	Jl. Mawar	4.1
23	YS	Jl. Sidomulyo Utara	0.6
24	YSu	Jl. Pahlawan	2.4
25	NNI	Jl. Rancang Sari	5.5

LAMPIRAN B DATA JARAK TIAP PENERIMA

Y	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17	Y18	Y19	Y20	Y21	Y22	Y23	Y24	Y25
Y0	0.0	3.5	2.2	2.7	6.2	2.0	1.7	1.1	4.5	3.3	5.1	3.0	2.5	1.8	1.3	2.3	6.0	4.6	3.6	2.9	0.8	7.4	4.1	0.6	2.4	5.5
Y1	3.5	0.0	4.4	1.0	6.9	2.1	2.9	2.4	3.4	4.5	3.7	3.8	3.1	4.0	2.4	1.4	6.5	2.2	0.7	3.9	9.2	8.0	5.6	2.6	1.4	5.1
Y2	2.2	4.4	0.0	1.5	5.6	3.0	3.1	8.2	4.4	3.9	2.8	5.2	2.9	2.2	2.7	1.9	3.0	2.6	1.1	2.0	8.4	7.1	3.3	2.9	1.8	4.2
Y3	2.7	1.0	1.5	0.0	7.2	1.8	3.9	2.2	6.3	4.0	3.2	1.1	4.6	3.6	3.0	4.7	2.2	0.6	9.9	3.1	11.1	2.5	3.8	4.0	8.0	7.6
Y4	6.2	6.9	5.6	7.2	0.0	6.7	1.2	11.0	2.7	5.2	3.9	6.0	7.8	5.4	6.7	4.4	1.8	8.3	2.2	4.2	2.4	3.4	0.9	5.3	1.1	9.0
Y5	2.0	2.1	3.0	1.8	6.7	0.0	4.8	9.3	5.6	9.7	2.1	0.6	1.5	0.9	8.2	7.4	0.6	3.9	8.4	5.6	6.0	0.3	7.0	6.2	7.6	3.2
Y6	1.7	2.9	3.1	3.9	1.2	4.8	0.0	0.5	9.0	2.2	0.9	2.7	0.4	6.7	12.3	8.3	9.9	13.6	4.4	14.2	7.4	6.2	8.1	2.4	0.4	2.7
Y7	1.1	2.4	8.2	2.2	11.0	9.3	0.5	0.0	4.8	1.7	9.4	3.2	13.9	1.3	10.8	12.6	12.0	7.3	11.4	9.8	1.4	1.7	10.3	8.8	12.6	0.5
Y8	4.5	3.4	4.4	6.3	2.7	5.6	9.0	4.8	0.0	6.1	5.2	1.1	11.6	13.2	3.1	6.3	4.9	2.2	8.0	0.5	2.2	4.5	14.1	11.3	3.5	5.0
Y9	3.3	4.5	3.9	4.0	5.2	9.7	2.2	1.7	6.1	0.0	3.4	12.0	5.0	2.3	13.3	10.1	0.5	1.9	13.9	11.2	14.2	0.7	5.7	12.6	10.2	1.5
Y10	5.1	3.7	2.8	3.2	3.9	2.1	0.9	9.4	5.2	3.4	0.0	10.0	14.9	11.9	3.3	1.3	12.1	3.1	4.1	10.6	2.4	4.7	3.5	6.0	14.7	4.8
Y11	3.0	3.8	5.2	1.1	6.0	0.6	2.7	3.2	1.1	12.0	10.0	0.0	3.2	3.8	15.0	11.8	4.6	13.4	10.7	2.3	11.9	12.7	14.5	2.5	0.5	5.6
Y12	2.5	3.1	2.9	4.6	7.8	1.5	0.4	13.9	11.6	5.0	14.9	3.2	0.0	8.3	1.2	0.7	14.0	5.1	14.8	3.8	5.4	11.7	4.2	8.1	14.5	4.9
Y13	1.8	4.0	2.2	3.6	5.4	0.9	6.7	1.3	13.2	2.3	11.9	3.8	8.3	0.0	6.0	13.5	2.4	10.8	12.2	14.3	3.7	4.6	12.8	3.6	0.7	7.0
Y14	1.3	2.4	2.7	3.0	6.7	8.2	12.3	10.8	3.1	13.3	3.3	15.0	1.2	6.0	0.0	4.2	0.9	11.3	0.6	1.4	10.9	13.8	2.0	2.6	4.6	3.2
Y15	2.3	1.4	1.9	4.7	4.4	7.4	8.3	12.6	6.3	10.1	1.3	11.8	0.7	13.5	4.2	0.0	9.0	2.5	5.5	13.6	5.9	3.3	14.4	11.0	9.2	11.6
Y16	6.0	6.5	3.0	2.2	1.8	0.6	9.9	12.0	4.9	0.5	12.1	4.6	14.0	2.4	0.9	9.0	0.0	5.3	12.3	3.8	8.5	2.6	11.0	11.4	3.7	3.0
Y17	4.6	2.2	2.6	0.6	8.3	3.9	13.6	7.3	2.2	1.9	3.1	13.4	5.1	10.8	11.3	2.5	5.3	0.0	4.0	12.9	3.4	4.3	5.6	1.6	5.8	11.8
Y18	3.6	0.7	1.1	9.9	2.2	8.4	4.4	11.4	8.0	13.9	4.1	10.7	14.8	12.2	0.6	5.5	12.3	4.0	0.0	7.6	13.0	2.8	11.3	4.3	3.4	9.0
Y19	2.9	3.9	2.0	3.1	4.2	5.6	14.2	9.8	0.5	11.2	10.6	2.3	3.8	14.3	1.4	13.6	3.8	12.9	7.6	0.0	15.0	3.9	12.4	12.0	11.7	13.1
Y20	0.8	9.2	8.4	11.1	2.4	6.0	7.4	1.4	2.2	14.2	2.4	11.9	5.4	3.7	10.9	5.9	8.5	3.4	13.0	15.0	0.0	13.7	2.7	3.9	4.8	11.4
Y21	7.4	8.0	7.1	2.5	3.4	0.3	6.2	1.7	4.5	0.7	4.7	12.7	11.7	4.6	13.8	3.3	2.6	4.3	2.8	3.9	13.7	0.0	11.6	4.4	12.5	2.9
Y22	4.1	5.6	3.3	3.8	0.9	7.0	8.1	10.3	14.1	5.7	3.5	14.5	4.2	12.8	2.0	14.4	11.0	5.6	11.3	12.4	2.7	11.6	0.0	1.7	11.5	1.0
Y23	0.6	2.6	2.9	4.0	5.3	6.2	2.4	8.8	11.3	12.6	6.0	2.5	8.1	3.6	2.6	11.0	11.4	1.6	4.3	12.0	3.9	4.4	1.7	0.0	5.8	4.5
Y24	2.4	1.4	1.8	8.0	1.1	7.6	0.4	12.6	3.5	10.2	14.7	0.5	14.5	0.7	4.6	9.2	3.7	5.8	3.4	11.7	4.8	12.5	11.5	5.8	0.0	1.8
Y25	5.5	5.1	4.2	7.6	9.0	3.2	2.7	0.5	5.0	1.5	4.8	5.6	4.9	7.0	3.2	11.6	3.0	11.8	9.0	13.1	11.4	2.9	1.0	4.5	1.8	0.0

LAMPIRAN C PERHITUNGAN NILAI FITNESS

P	Kromosom															Total jarak	Fitness
	Urutan Pengiriman																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
P1	3	11	7	1	13	10	5	14	8	12	4	2	15	9	6	76.2	13.123
		1.1	3.2	1.1	4.0	11.9	2.1	8.2	3.1	11.6	7.8	6.9	2.9	10.1	2.2		
P2	5	2	9	15	11	6	13	1	7	14	3	12	8	10	4	83.7	11.947
		3.0	3.9	10.1	11.8	2.7	6.7	4.0	2.4	10.8	3.0	4.6	11.6	5.2	3.9		
P3	7	4	6	11	13	8	2	5	10	12	1	3	15	9	14	88.5	11.299
		11.0	1.2	2.7	3.8	13.2	4.4	3.0	2.1	14.9	3.1	1.0	4.7	10.1	13.3		
P4	2	9	12	4	15	7	10	3	14	8	5	11	7	1	13	68.2	14.662
		3.9	5.0	7.8	4.4	12.6	9.4	3.2	3.0	3.1	5.6	0.6	3.2	2.4	4.0		
P5	12	4	14	11	8	13	7	10	1	15	6	9	3	5	2	78.9	12.674
		7.8	6.7	15.0	1.1	13.2	1.3	9.4	3.7	1.4	8.3	2.2	4.0	1.8	3.0		
C1	3	11	7	1	13	10	5	2	9	15	6	14	12	8	4	77.8	12.853
		1.1	3.2	2.4	4.0	11.9	2.1	3.0	3.9	10.1	8.3	12.3	1.2	11.6	2.7		
C2	5	2	9	15	11	6	13	3	7	1	10	14	8	12	4	75.9	13.175
		3.0	3.9	10.1	11.8	2.7	6.7	3.6	2.2	2.4	3.7	3.3	3.1	11.6	7.8		
C3	2	9	12	11	15	7	10	3	14	8	5	4	7	1	13	96.6	10.351
		3.9	5.0	14.9	11.8	12.6	9.4	3.2	3.0	3.1	5.6	6.7	11.0	2.4	4.0		

LAMPIRAN D UJI COBA UKURAN POPULASI

Ukuran Populasi	Nilai Fitness Percobaan ke - n										Fitness Tertinggi	Rata – Rata Fitness
	1	2	3	4	5	6	7	8	9	10		
10	0,08753	0,08885	0,09681	0,09814	0,10079	0,10079	0,10212	0,10477	0,10610	0,11405	0,11405	0,090178
20	0,09012	0,09227	0,09298	0,09656	0,09942	0,10085	0,10228	0,10301	0,10515	0,11731	0,11731	0,100281
30	0,08061	0,09183	0,09336	0,09693	0,09795	0,09846	0,10511	0,10969	0,11021	0,11581	0,11581	0,099996
40	0,09171	0,09317	0,09647	0,09757	0,09867	0,09977	0,10051	0,10308	0,10344	0,11555	0,11555	0,099994
50	0,09247	0,09803	0,09803	0,09833	0,09862	0,10067	0,10301	0,10301	0,10359	0,10418	0,10418	0,099994
60	0,09286	0,09669	0,09746	0,09797	0,09849	0,10002	0,10028	0,10156	0,10591	0,10872	0,10872	0,099996
70	0,09094	0,09367	0,09786	0,09807	0,09932	0,10121	0,10311	0,10311	0,10331	0,10938	0,10938	0,099998
80	0,09559	0,09559	0,09711	0,09731	0,09806	0,09972	0,10053	0,10261	0,10603	0,10735	0,10735	0,18979
90	0,09641	0,09792	0,09792	0,09809	0,09911	0,09927	0,10062	0,10265	0,10332	0,10467	0,10467	0,100079
100	0,09381	0,09758	0,09899	0,09931	0,10103	0,10103	0,10151	0,10181	0,10213	0,10276	0,10276	0,099996

LAMPIRAN E UJI COBA BANYAK GENERASI

Banyak Generasi	Nilai Fitness Percobaan ke - n										Fitness tertinggi	Rata – rata fitness
	1	2	3	4	5	6	7	8	9	10		
50	0,09546	0,09607	0,09728	0,09924	0,99395	0,09984	0,10015	0,10091	0,10438	0,10725	0,10725	0.189453
100	0,09248	0,09478	0,09555	0,09571	0,09984	0,10015	0,10306	0,10337	0,10691	0,10812	0,10812	0.099997
150	0,09523	0,09774	0,09853	0,09869	0,09869	0,10058	0,10073	0,10278	0,10325	0,10373	0,10373	0.099995
200	0,09386	0,09751	0,09781	0,09857	0,10024	0,10024	0,10024	0,10221	0,10343	0,10586	0,10586	0.099997
250	0,09498	0,09789	0,09805	0,09821	0,09897	0,09989	0,10035	0,10251	0,10372	0,10541	0,10541	0.099998
300	0,09565	0,09611	0,09751	0,09875	0,09922	0,09937	0,10217	0,10311	0,10357	0,10451	0,10451	0.099997
350	0,09447	0,09602	0,09664	0,09911	0,10051	0,10128	0,10159	0,10236	0,10361	0,10437	0,10437	0.099996
400	0,09737	0,09807	0,09891	0,09948	0,10018	0,10061	0,10074	0,10088	0,10158	0,10214	0,10214	0.099996
450	0,08974	0,09303	0,09350	0,09835	0,10023	0,10352	0,10383	0,10477	0,10493	0,10806	0,10806	0.099996
500	0,09202	0,09564	0,09815	0,09878	0,09911	0,09973	0,10114	0,10366	0,10429	0,10744	0,10744	0.099996

LAMPIRAN F UJI COBA KOMBINASI CR & MR

Kombinasi		Nilai Fitness Percobaan ke - n										Fitness tertinggi	Rata – rata fitness
Cr	Mr	1	2	3	4	5	6	7	8	9	10		
0	1	0,09391	0,09557	0,09633	0,09861	0,09875	0,10133	0,10209	0,10345	0,10496	0,10496	0,10496	0,099996
0.1	0.9	0,09597	0,09663	0,09793	0,09811	0,09811	0,10121	0,10202	0,10284	0,10317	0,10398	0,10398	0,099997
0.2	0.8	0,09281	0,09541	0,09971	0,09986	0,10016	0,10017	0,10078	0,10185	0,10231	0,10691	0,10691	0,099997
0.3	0.7	0,09673	0,09688	0,09749	0,09749	0,09839	0,09870	0,09991	0,10459	0,10474	0,10504	0,10504	0,099996
0.4	0.6	0,09428	0,09584	0,09662	0,09849	0,09913	0,09957	0,10238	0,10301	0,10471	0,10595	0,10595	0,099998
0.5	0.5	0,09157	0,09471	0,09595	0,09829	0,09876	0,10142	0,10189	0,10345	0,10564	0,10829	0,10829	0,100933
0.6	0.4	0,09247	0,09673	0,09704	0,09752	0,09831	0,10178	0,10241	0,10272	0,10272	0,10825	0,10825	0,099995
0.7	0.3	0,09579	0,09809	0,09809	0,09871	0,09946	0,10038	0,10038	0,10114	0,10175	0,10618	0,10618	0,099997
0.8	0.2	0,09557	0,09608	0,09780	0,09898	0,09928	0,09973	0,10166	0,10271	0,10329	0,10404	0,10404	0,099909
0.9	0.1	0,09338	0,09663	0,09825	0,09841	0,10047	0,10165	0,10165	0,10181	0,10357	0,10416	0,10416	0,099998
1	0	0,09581	0,09581	0,09918	0,09934	0,09966	0,10078	0,10094	0,10094	0,10335	0,10415	0,10415	0,099996